

ARI Research Note 89-32

AD-A219 967

Judging Samples of Command Post and Field Training Exercise Messages

Robert E. Solick
U.S. Army Research Institute

and

Michael J. Liebhaber, Richard W. Obermayer,
James M. Linville, and Andrew H. Obermayer
Vreuls Research Corporation

for

Contracting Officer's Representative
Robert E. Solick

Field Unit at Fort Leavenworth, Kansas
Stanley M. Halpin, Chief

Systems Research Laboratory
Robin L. Keesee, Director

June 1989



United States Army
Research Institute for the Behavioral and Social Sciences

Approved for the public release; distribution is unlimited.



90 04 02 077

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

**A Field Operating Agency Under the Jurisdiction
of the Deputy Chief of Staff for Personnel**

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Research accomplished under contract
for the Department of the Army

Vreuls Research Corporation

Technical review by

Lloyd M. Crumley

NOTICES

DISTRIBUTION: This report has been cleared for release to the Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or the National Technical Information Service (NTIS).

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The views, opinions, and findings in this report are those of the author(s) and should not to be construed as an official Department of the Army position, policy, or decision, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS --		
2a. SECURITY CLASSIFICATION AUTHORITY --			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE --			4. PERFORMING ORGANIZATION REPORT NUMBER(S) VRC-TR-8801		
6a. NAME OF PERFORMING ORGANIZATION Vreuls Research Corporation			6b. OFFICE SYMBOL (If applicable) --		5. MONITORING ORGANIZATION REPORT NUMBER(S) ARI Research Note 89-32
6c. ADDRESS (City, State, and ZIP Code) 68 Long Court, Suite E Thousand Oaks, CA 91360			7a. NAME OF MONITORING ORGANIZATION Army Research Institute Fort Leavenworth Field Unit		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences			8b. OFFICE SYMBOL (If applicable) PERI-S		7b. ADDRESS (City, State, and ZIP Code) P.O. Box 3407 Ft. Leavenworth, KS 66027-0347
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600			9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-86-C-0210		
11. TITLE (Include Security Classification) Judging Samples of Command Post and Field Training Exercise Messages			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO. 62785A	PROJECT NO. 791	TASK NO. 1301
12. PERSONAL AUTHOR(S) Solick, Robert E. (ARI); Liebhaber, Michael J.; Obermayer, Richard W.; Linville, James M.; Obermayer, Andrew H. (Vreuls Research Corp.)					
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 87/03 TO 87/11		14. DATE OF REPORT (Year, Month, Day) 1989, June		15. PAGE COUNT 90
16. SUPPLEMENTARY NOTATION Robert E. Solick, Contracting Officer's Representative					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Field training exercises Semantic scales Messages		
			Command post exercises Expert panel Research		
			Bi-polar adjectives Training		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>➤ This report documents the performance of a computer-administered measurement instrument. This instrument, MsgJudge, presents messages from two environments to a panel of expert judges and collects data in the form of responses to semantic scales, forced-choice questions, and a summary questionnaire.</p> <p>Radio transmissions from a Battalion Command Net and from a Task Force Command Net were transcribed. Transcription included removal of call signs and most procedural words. The individual messages were then presented to experts, who were asked to rate them on a scale with bi-polar adjectives at either end. These adjectives were selected to allow the expert to rate the message in the areas of evaluation, potency, activity, and style. The rater was then asked to decide whether the message came from a Field Training Exercise (FTX) or from a Command Post Exercise (CPX).</p> <p style="text-align: right;">(Continued)</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Robert E. Solick			22b. TELEPHONE (Include Area Code) (913) 684-4933		22c. OFFICE SYMBOL PERI-SL

ARI Research Note 89-32

19. ABSTRACT (Continued)

The raters chose correctly between FTX and CPX when presented with that forced choice. Use of the bi-polar adjectives on a semantic differential scale also indicated the correct identification of the source.

This report presents the data collected in a pilot test of the instrument. The data will be used to select experimental parameters for later use of the instrument to examine the factors that distinguish command post exercises from field exercise messages. The pilot results indicate that both the number of response scales and the number of messages may be reduced in subsequent experiments without reducing the power of the instrument.



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

JUDGING SAMPLES OF COMMAND POST AND FIELD TRAINING EXERCISE MESSAGES

CONTENTS

	Page
INTRODUCTION	1
DESCRIPTION OF THE INSTRUMENT	1
The Bi-Polar Adjective Approach	2
Measurement Implementation	4
Questionnaire	5
PILOT TEST	5
Design	5
Analysis	5
Summary	6
ANALYSIS OF THE INSTRUMENT	6
Discriminant Analysis	7
Cluster Analysis	8
Reliability	8
EFFICIENCY CRITERIA	8
Selection of the Most Discriminating Scales	8
Reduction of the Total Number of Messages Administered	9
CONCLUSIONS	9
REFERENCES	11
APPENDIX A. MSGPREP PROGRAM	A-1
B. MSGJUDGE PROGRAM	B-1
C. QUESTIONS ADMINISTERED TO PANEL	C-1
D. MESSAGES USED IN MSGJUDGE	D-1
E. INSTRUCTIONS TO PANEL	E-1
F. SOURCE TABLE FROM ANALYSIS	F-1
G. PRELIMINARY ANALYSIS RESULTS	G-1

CONTENTS (Continued)

Page

LIST OF TABLES

Table 1. Semantic differential scales	3
2. Matrix of Spearman correlation coefficients for the semantic differential (SD) scales	7
3. Discriminant contingency table for observed and predicted message assignment	7
4. Reduced semantic differential scales	9

JUDGING SAMPLES OF COMMAND POST AND FIELD TRAINING EXERCISE MESSAGES

Introduction

MsgJudge is a computer-based measurement instrument that administers messages from two environments to a panel of expert judges and collects data in the form of responses to semantic scales, forced-choice questions, and a summary questionnaire. The purpose of this report is to document the instrument and to present analyses of its performance based on a preliminary small-scale test.

The research problem for which MsgJudge was created stems from perceived differences in communications between Field Training Exercises (FTX) and Command Post Exercises (CPX) and the potential for a consequent training differential. The message traffic for comparison was taken from an ARTBASS exercise and a National Training Center rotation.

In the CPX environment the company level activities and actions are reported to the battalion staff via the Battalion Command Net by role players who take input directly from the simulation, create a tactical message, and then transmit the information to the battalion commander and staff. If this created message presents information in a manner recognizably different from that presented in actual radio transmissions from the company level in an FTX, the actions, responses, and, more importantly, the training received at the battalion level may be affected. Of concern is the possible degradation of transfer of training when moving from the CPX to the FTX environment.

The initial research objective was to determine if an expert panel can identify differences in messages by examining the message transmission out of context. Through use of a rating scheme with bi-polar adjectives, the panel was asked to rate the transmission in terms of evaluation, potency, activity and style. The panel member was also asked to indicate if the message comes from an FTX or CPX environment. This rating activity was administered through a computer program, MsgJudge, that randomly presents messages from both environments, provides a rating scale using multiple discriminators, and provides for compilation of data.

This report contains a description of the instrument, a summary of a pilot test of the instrument, and appendixes that present the computer programs and the set of messages used in the pilot test.

Description of the Instrument

Each expert panel member is presented with a series of messages randomly selected from the FTX and CPX environment. (A computer program ensures that an equal number of messages from both environments are presented and that selected categories of messages are sampled.) The panel member is also presented with a bi-polar rating scale (also randomly selected, but each scale from each of the rating areas is eventually displayed) and the question "Does this transmission

come from an FTX or a CPX?" Analysis of the responses can determine whether CPX and FTX messages are recognizable by the panel, and can statistically describe CPX and FTX messages in terms of style, evaluation, potency, and activity.

The Bi-Polar Adjective Scale Approach

The research required a simple and inexpensive instrument that can be used to measure the affective responses of military subject matter experts when they are exposed to samples of the test messages. There is a well-established method that seems to meet the requirements, called the Semantic Differential (Osgood, Suci, & Tannenbaum, 1957; Heise, 1970).

The Semantic Differential measures reactions to stimulus entities (words, concepts, messages, pictures, personalities, etc.) in terms of ratings on bi-polar scales, e.g.:

GOOD : : : : : : : : BAD

Some of the characteristics of the Semantic Differential are:

- Bi-polar adjective scales are a simple and economical means for obtaining data on people's reactions.
- Ratings on bipolar adjective scales tend to be correlated, and factor analysis indicates there are at least three major stable dimensions labeled Evaluation, Potency, and Activity. This result has been replicated in many studies.
- Measurements on these three dimensions are appropriate for affective responses, applicable to any concept or stimulus.
- The Semantic Differential has been used as a measure of affective response in a wide variety of projects which support its validity as a measurement technique. For example, it was used successfully in measuring attitudes toward person-person communication media (Champness, 1973) and as an index of traffic sign perception and comprehension (Dewar & Ells, 1977).

In designing an instrument based on the bi-polar adjective approach, it is desirable to select adjectives which have been demonstrated to be bi-polar, and which correlate exclusively with one of the stable underlying dimensions. Using these criteria, the following bi-polar adjectives were selected as candidates from factor analysis data in the literature (Osgood, 1952; Carroll, 1959; Osgood, 1964; Osgood, & Luria, 1954; Friedman, Johnson, & Fode, 1964; Osgood, 1959; Mirion, 1961):

- Evaluative: good-bad; valuable-worthless; useful-useless; familiar-unfamiliar
- Potency: powerful-powerless; long-short; strong-weak; deep-shallow
- Activity: active-passive; fast-slow; excitement-calm; energetic-lazy

It should be noted that some factor analytic studies of the Semantic Differential have found more than one evaluative factor (e.g., Snyder & Wiggins, 1970). Consequently, the work of Carroll (1960) on the application of the method to stylistic elements in writing was examined with interest, and the following bi-polar adjective candidates were extracted as being relevant to military messages:

- Style: meaningful-meaningless; succinct-wordy; ordered-chaotic; precise-vague

It is possible that some or all of these bi-polar scales will correlate with the Evaluative scales above; however, that is a matter that can be decided by empirical test. Each of the bi-polar scales derived from the Carroll test battery conform to external considerations of message efficacy. Based on the foregoing considerations, the bi-polar adjective scales in Table 1 have been selected for initial testing. The first three are selected from the standard dimensions, while the remainder are selected from the Carroll Style scales. All except the strong-weak and active-passive scales relate to evaluation. The evaluative scales are potentially different, but of course may prove to be either nondiscriminating or to correlate with the other scales. This issue is addressed in the analysis below.

Table 1

Semantic Differential Scales

<u>Scale</u>	<u>Adjective</u>	
1	valuable	-- worthless
2	strong	-- weak
3	active	-- passive
4	meaningful	-- meaningless
5	succinct	-- wordy
6	ordered	-- chaotic
7	precise	-- vague

In addition to the Semantic Differential scales, the initial test included a forced-choice judgment as to the source of a stimulus message (either CPX or FTX), and a summary questionnaire.

Measurement Implementation

The measurement was implemented on a personal computer, with a stimulus message displayed across the top of the screen, and each of the seven bi-polar adjective scales successively occupying the space below. For a given message, all of the scales were administered one at a time, until all measurement was completed for that question. The forced-choice question (CPX or FTX) was also presented for each message, but only after all messages were presented with bi-polar adjective scales. A typical screen is shown below:

Message: Roger. I know they aren't supposed
to be using them. My guys reported a motor-
cycle. Grid 585 125.

Valuable : : : : : : : : Worthless
pointer

A pointer was movable by the judge in small increments and the pointer location was automatically recorded. The next screen showed the same message with a different scale, and this was repeated until data from all seven scales were collected. Since the judge's response could be affected by order of message presentation, scale presentation, orientation of the scale and the initial location of the pointer, all of these factors were randomized by the computer program.

The computer program was divided into two parts: MsgPrep creates a random selection of N messages for subsequent use, MsgJudge automatically conducted the remainder of the data collection using control files generated by MsgPrep. These programs are in Appendices A and B respectively.

The MsgPrep program must be executed prior to all testing to set the conditions desired for all testing. During the execution of MsgPrep the following parameters are defined through interactive dialogue:

- Number of semantic differential scales
- Number of pages of instructions for use of MsgJudge
- Number of messages
- Number of message categories
- Number of equal sessions of messages
- Seed number for randomization
- Same random order to be repeated for each expert (yes or no)

- Semantic differential scales to be included in test (yes or no)
- Forced-choice question as to FTX or CPX (after all semantic differential scales, before semantic differential scales for each message, or no forced-choice question at all)
- Questionnaire to be included (yes or no)
- Codes to be used to identify types of data to be recorded

Once MsgPrep has been executed, the message and control files for MsgJudge are created, and the instrument is ready for use.

Questionnaire

At the end of the experimental session a questionnaire, shown in Appendix C, was computer-administered. The questionnaire asks for opinions about the characteristics that distinguish CPX from FTX messages.

Pilot Test

The instrument was administered to three Army officers, each of whom were experienced in both CPX and FTX environments. The set of messages consisted of transcriptions of tactical communications recorded during comparable CPX and FTX exercises. Each message set was subcategorized according to topic: enemy sightings and information, or operational information about one's own forces. Two additional categories of information were recorded during the exercises: administrative messages and command guidance. These categories were omitted from the pilot test due to inadequate sample size. The experimental message set (presented in Appendix D) consisted of 73 friendly information and 53 enemy information messages from the FTX and 87 friendly and 80 enemy messages from the CPX. The MsgPrep program randomly selected 15 messages from each category, or 60 messages in all, to be judged by each officer.

The instrument was administered in the afternoon of 15 July 1987 to Army officers serving at the Naval Amphibious School. Computer-administered instructions are shown in Appendix E. The program was administered twice to one officer to obtain preliminary information on test-retest reliability.

Design

A 2X2 factorial, within subjects design was used. The independent variables were message source (CPX or FTX) and message topic (enemy or friendly information). Dependent variables consisted of responses on each of the seven semantic differential scales and of a forced-choice response as to the origin of each message (CPX or FTX).

Analysis

An analysis of variance was performed to determine whether the officers differed in their average ratings on the semantic differential. There was no

subject effect ($F(2, 177) = 1.0, p > .90$), but subjects did respond differentially on the seven scales ($F(12, 1062) = 2.81, p < .001$). An ANOVA using forced choice responses as the dependent measure also revealed no differences among subjects, ($F(2, 177) = .20, p > .80$). ANOVA source tables for those analyses are given in Appendix F. Subsequent analyses pooled data across subjects.

A 2(message source) x 2 (message topic) repeated measures ANOVA was performed on the messages. Average responses on the semantic differential scales are given in Appendix G. There was an effect for message topic ($F(1, 176) = 26.16, p < .001$) but not for source ($F(1, 176) = 3.30, p > .05$), nor for the source by topic interaction ($F(1, 176) = .02, p > .80$).

An analysis of the forced choice data revealed that performance was significantly better than chance (chi-square (1) = 24.4, $p < .001$), but significantly worse than perfect (chi-square (3) = 18.5, $p < .005$). Overall, 63.3% of the CPX messages and 73.3% of the FTX messages were correctly classified as to their origins.

An estimate of the sample size required to obtain results within 1% of the true mean was computed using the method recommended by Woods, Fletcher and Hughes (1986). A sample of 19 subjects would provide this degree of precision.

Summary

The analysis indicated that the officers could discriminate among various types of tactical radio messages, and that message topic and message source are relevant factors in that discrimination. The officers could also identify the source better than chance selection, suggesting that the desired level of similarity in training environments has not been reached.

Analysis of the Instrument

The above results indicate that experienced subjects differentially rate messages based on the message content and, to a lesser extent, source. They can also correctly identify the source of the message. The following analysis was done to determine how well the Semantic Differential (SD) scales performed. That is, how well did an individual scale predict message source and content, and did any scales overlap? The analysis was done in three stages. First was a discriminant analysis to assess the a priori selection of the seven scales as valid predictors of message category. This was followed by some exploratory analyses to discover natural groupings in the data. The final stage was an examination of test-retest reliability. Table 2 shows the correlations among the SD scales. Given the small sample size, the results of these analyses should be viewed as tentative and exploratory.

Table 2

Matrix of Spearman Correlation Coefficients for the Semantic Differential (SD) Scales

	SD(1)	SD(2)	SD(3)	SD(4)	SD(5)	SD(6)	SD(7)
SD(1)	1.000						
SD(2)	0.568	1.000					
SD(3)	0.423	0.604	1.000				
SD(4)	0.763	0.604	0.469	1.000			
SD(5)	0.087	0.019	-0.071	0.188	1.000		
SD(6)	0.109	0.035	-0.040	0.220	0.667	1.000	
SD(7)	0.553	0.418	0.234	0.533	0.369	0.393	1.000

Discriminant Analysis

A discriminant analysis was done to assess the validity of the current set of scales. If the seven scales are valid, they should predict message source and content a significant amount of the time. The source and content data were grouped into four categories: CPX-E, CPX-F, FTX-E, and FTX-F. A one-way MANOVA on the scales indicated that the categories were relatively different from each other (Wilk's Lambda = .617, $p < .001$). The discriminant functions are:

$$\text{Factor(1)} = -0.924*SD(1) + 0.546*SD(2) + 0.160*SD(3) + 0.085*SD(4) + \\ -0.253*SD(5) + 0.298*SD(6) + -0.587*SD(7)$$

$$\text{Factor(2)} = 0.099*SD(1) + 0.770*SD(2) + -0.199*SD(3) + 0.316*SD(4) + \\ -0.604*SD(5) + 0.906*SD(6) + -0.432*SD(7)$$

$$\text{Factor(3)} = 0.207*SD(1) + -0.087*SD(2) + 0.405*SD(3) + -0.822*SD(4) + \\ 0.201*SD(5) + 0.789*SD(6) + 0.123*SD(7)$$

The observed and predicted assignment of messages to categories are in Table 3.

Table 3

Discriminant Contingency Table for Observed and Predicted Message Assignment

Actually in Category	Predicted to be in Category				Total
	CPX-E	CPX-F	FTX-E	FTX-F	
CPX-E	21	6	12	6	45
CPX-F	8	18	10	9	45
FTX-E	12	8	20	5	45
FTX-F	6	7	5	27	45
Total	47	39	47	47	180

The scales were successful at predicting message category (Pearson Chi-Sq(9) = 57.88, $p < .001$). For example, of the 45 messages that were actually in the CPX-E category, 21 were correctly assigned. The remaining 24 were incorrectly assigned to CPX-F (6), FTX-E (12), and FTX-F (6).

The discriminant analysis revealed that the SD scales were able to predict message type. However, it was far from perfect. This could have been caused by high correlations among different SD scales. A Cluster Analysis was done to study the correlations and to determine if one or more scales could be combined or dropped.

Cluster Analysis

A hierarchical Cluster Analysis revealed the natural groupings of the seven SD scales. Scales 5 and 6 cluster together as do 1 and 4 and the group of 1, 2, 3, 4, and 7.

Reliability

One officer was retested to measure test-retest reliability. One-way ANOVAs were performed on the Differential responses and on the Forced Choice responses with Test as the factor. There was no significant difference in the subject's responses from test one to test two for either response type. For the Differential responses, $F(1, 118) = 2.66$, $p > .10$, and $F(1, 118) = .30$, $p > .50$ for the Forced Choice responses.

Efficiency Criteria

Discriminant Analysis indicated that the seven SD scales could successfully categorize the radio messages. However, the fit was not perfect. Since rating a large number of messages on seven scales is time consuming, fewer scales and messages would be preferable, and could also increase the discriminatory power of the message analysis method.

Selection of the Most Discriminating Scales

The analysis revealed that a reduction in the number of scales is possible since some scales tend to cluster (1-2-3-4-7 & 5-6). The three dimensions shown in Table 4 should be able to provide a level of discrimination equal to or greater than the seven used in the pilot study. The revised scales are reduced from correlated groups that were identified in the Cluster Analysis. Scale SD(3) (active-passive) was excluded since it is apparently a poor discriminator among the different categories of messages. This decision was based on its failure to achieve significance in the Discriminant Analysis.

Table 4

Reduced Semantic Differential Scales

<u>Reduced Scale</u>	<u>Collapsed from Scales</u>
ordered - chaotic -----	SD(5) succinct - wordy SD(6) ordered - chaotic
valuable - worthless _____	SD(1) valuable - worthless SD(4) meaningful - meaningless
precise - vague _____	SD(2) strong - weak SD(7) precise - vague

An analysis was done on the reduced SD scales in Table 4 to ensure that this subset could perform as well, or better than, the entire set of scales. Use of the three scales SD(1), SD(6) and SD(7) instead of the entire set of seven scales yielded discriminant functions that could predict message source as well as the entire set. Therefore, it appears that substantial savings in time required of expert judges may be achieved by reducing the number of scales.

Reduction of the Total Number of Messages Administered

Fewer messages within each group (CPX, FTX, friendly, enemy) would further reduce the required time to administer the MsgJudge program. The number of required messages is estimated (cf. Kirk, 1968:142) from the power of the analysis. The calculated power, with the available number of messages, is much greater than 0.99 (i.e., power = $1 - \beta \gg .99$). There are 45 messages in each group, and since the power is high and fewer messages would be desirable, a new sample size, N, was computed. The new size, with a priori values of power = .95 and alpha = .05, is $N = 17.6$ messages. Therefore, the number of messages in each group could be reduced to this value without an appreciable loss of power.

Conclusions

The two goals of this preliminary study have been accomplished. First, a survey was created and successfully administered. An analysis of the results indicated that the expert subjects rated messages differently depending on the source of the messages. Second, an analysis of the differential scales used in the survey indicated that some were better than others at predicting the source and content of individual messages. However, the scales performed well overall and the results appeared to be stable from one test administration to the next. Additional analysis suggested that the number of semantic differential scales and messages should be reduced. This reduction would result in a concise, accurate, and easily administered survey. Therefore, the MsgJudge program, in a revised form with three semantic differential scales and 20 messages, appears to be a satisfactory vehicle to survey an expert panel on message differences.

Comments of the officers, at the conclusion of the program, indicated some excitement about the project, some concern about the amount of time required to complete (1 hour; however the analysis indicates the potential to reduce this amount of time), and a real interest in the deeper question of how the results of the study will relate to training. The reduction of the number of semantic differential scales is expected to decrease the time of the survey. Projected training benefits include better modeling of FTX messages in CPX simulations so as to provide a more realistic training environment than is currently available. Additional use of the MsgJudge survey to query experts on communication differences will ensue.

References

- Carroll, J. B. (1959). Review of the measurement of meaning. Language, 35, 58-77.
- Carroll, J. B. (1960). Vectors of prose style. In Sebeok, T. A. (Ed.), Style in Language. New York: John Wiley & Sons.
- Champness, B. G. (1973). Attitudes toward person-person communications media. Human Factors, 15, 437-447.
- Dewar, R. E., & Ells, J. G. (1977). The semantic differential as an index of traffic sign perception and comprehension. Human Factors, 19, 183-189.
- Friedman, C. J., Johnson, C. A., & Fode, K. (1964). Subject's descriptions of selected TAT cards via the semantic differential. Journal of Consulting Psychology, 28, 317-325.
- Heise, D. R. (1970). The semantic differential and attitude research. In Summers, G. F. (Ed.), Attitude measurement. Chicago: Rand McNally.
- Heise, D. R. (1982, July). Measuring attitudes with a PET: A BASIC program that finds out how people feel. Byte, 208-246.
- Kirk, R. E. (1968). Experimental design procedures for the behavioral sciences. Belmont, CA: Brooks/Cole.
- Mirion, M. S. (1961). A cross-linguistic investigation of phonetic symbolism. Journal of Abnormal & Social Psychology, 62, 623-630.
- Osgood, C. E. (1952). The nature and measurement of meaning. Psychological Bulletin, 49, 197-237.
- Osgood, C. E. (1959). The cross-cultural generality of visual-verbal synesthetic tendencies. Behavioral Science, 5, 146-169.
- Osgood, C. E., & Luria, Z. (1954). A blind analysis of a case of multiple personality using the semantic differential. Journal of Abnormal & Social Psychology, 49, 579-591.
- Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). The measurement of meaning. Urbana, IL: University of Illinois Press.
- Osgood, C. E. (1964). Semantic Differential Technique in the comparative study of cultures. American Anthropologist, 66, 171-200.
- Snyder, F. W., & Wiggins, N. (1970). Affective meaning systems: A multivariate approach. Multivariate Behavioral Research, 5, 453-468.

Summers, G. F. (1970). Attitude measurement. Chicago: Rand McNally & Co.

Woods, A., Fletcher, P., & Hughes, A. (1960). Statistics in language studies.
New York: Cambridge University Press.

APPENDIX A

MSGPREP PROGRAM

Below is the source code listing of the MSGPREP program. It is written in TurboPascal and runs on an IBM PC.

Program MsgPrep (Input,Output);

(*

Basic program & file flow for MsgPrep & MsgJudge:

<u>MsgPrep</u>		<u>MsgJudge</u>	
<u>Input Files</u>	<u>Output Files</u>	<u>Input Files</u>	<u>Output Files</u>
(1) ConFile.txt	ConFile.txt	====> ConFile.txt	ConFile.txt
(2)	PrepMsg.txt	====> PrepMsg.txt	
(3) Instruct.Doc	=====>	Instruct.Doc	
(4) DiffScal.Txt	=====>	DiffScal.Txt	
(5) Messages.Txt			
(6)			A data file
(7)			An info. file

The data file name will be as follows: Initials+Month+Day.Dat

The Info. file name will be as follows: Initials+Month+Day.Inf

The general purpose of MsgPrep is to do all the pre-processing that is required before the execution of MsgJudge. This pre-processing is needed each time a new setup is desired. The set-up will allow the researcher to specify: the number of messages, the type of questions to be given to the expert, and other information needs for MsgJudge. There are two kinds of questions that can be given: (1) A force-choice question which will ask the expert which environment he feels the message came from, and (2) a series of semantic differential scales for the expert to rate the message on. Either of these question types can be given separately or together. If both are given then they can be given in either order. After all questions have been given, a survey can be given if desired. This pre-processing step is designed to decrease the amount of time needed to run msgjudge.

These files are input files :

- (1) Instruct.Doc
- (2) ConFile.Txt
- (3) Messages.Txt
- (4) DiffScal.Txt

These files are outputs:

- (1) ConFile.Txt
- (2) PrepMsg.Txt

Any changes desired to the input files of MsgPrep, should be made before MsgPrep is run. If any of the input files are changed, MsgPrep must be re-run.

Description of each of the input files:

Instruct.Doc file is used in determining the number of pages in the file of instructions.

ConFile.txt is used so that new values don't have to be entered in MsgPrep. If it is desired to leave any entry value the same as the previous set-up, a carriage return may be entered for any of the questions. Warning!, the seed used for creating the random number changes after every execution of MsgJudge. If the same messages are desired the original seed must be used. It is recommended that the researcher write the seed used directly on the set-up disk.

Messages.Txt is the file from which the messages for MsgJudge are chosen and pulled.

DiffScal.Txt is needed to determine the number of differential adjective scales for ConFile.Txt. To modify any of these input files, use a text editor and then re-run MsgPrep.

MsgPrep has two output files, PrepMsg.Txt & ConFile.Txt. PrepMsg.Txt consist of the messages chosen randomly to be given to the subjects in MsgJudge. This file need not be present to run MsgPrep. ConFile.Txt is the control file for MsgJudge. This file is used to set-up the order in which types of questions are asked and whether or not they will be asked at all. For convenience of the researcher MsgPrep reads the old ConFile.Txt in and allows researcher to keep any of the old values. To keep old values just strike a carriage return when the input is requested. This feature makes it necessary to have a Confile.Txt present on the disk. If no ConFile is present, create one with a text editor, using the format given later in the File definition section of the documentation.

Basic program flow :

- (1) Determine information from input files for confile.
 - i. Read in old Confile, so that researcher can keep any old information.
 - ii. Determine the number of Scales being used. Value is determined by counting the number of lines in the DiffScal.Txt file. Note that blank lines may be misinterpreted.
 - iii. Determine the number of pages in the instructions. Number of pages is determined by finding the number of lines and dividing by 24. An extra page is added for any partial pages.
 - iv. Determine the number of categories and the number of messages in each of the categories. The number of categories are found by counting the number of occurrences of the string 'MsgCat' and the number of messages are found by counting the number of the five line groups between the MsgCat strings or between a MsgCat and the end of the file.

- (2) Ask if any information needs to be updated, which could not be automatically determined.
- (3) The messages are then randomly chosen, found and written to the Outfile. To do this the desired category is chosen, then a message from that category between 1 and the maximum number of messages is chosen. Then the infile (Messages.Txt) is reset. The category is searched for followed by a message number. Finally the message is written to the OutFile (PrepMsg.Txt). This is done once for each of the messages to be presented to the expert.
- (4) Finally, all the outfiles are closed and execution is halted.

VARIABLES DEFINITIONS:

FixOrder: This is a character that defines the order in which msgjudge presents the messages. If yes, the messages are given to all experts in the same random order found in the input file PrepMsg.Txt. If no, then the messages are randomized again by MsgJudge, and the experts will see the messages displayed in a different random order.

SurBool: This boolean will turn the Survey off and on in MsgJudge.

DifQuest: This character is used to turn on and off the differential question. If yes then the expert is given each message with the differential adjective scales. If no then the expert gets no differential scales and the forcequest variable is automatically defined as after.

Ch: Temporary Character variable.

MsgTypel: This variable is used for the force-choice question. It defines the first environment for the force-choice question.

MsgType2: This variable is also used for the force-choice question. It defines the second environment.

ForceQuest: This variable defines when, if at all, the force-choice question will be presented to the expert. The following are the choices for ForceQuest:

- (1) 'After' - this option will present the forced question after all the differential scales have been answered for all the messages.
- (2) 'Before' - this option will present the forced question before the differential scales for each of the messages. In other words, for each message the forced question is asked first then the differential scales are asked.
- (3) 'No'- no forced question to be asked.

Str80: This variable is used for reading a line of 80 characters from a file for display purposes. It is used in the survey section to display text and for writing messages.

MsgUsed: Is an array used to keep track of which messages have already been used. This is only important in the mode when the messages are not taken in the order that they are found in the file 'PrepMsg.Txt'. It should be noted that messages in 'PrepMsg.Txt' are in random order. The random order is used when you wish the message to be given in a different random order to each of the experts.

Seed: This variable is a real number greater than 0 and less than 1. It is used to determine the next random number. First, the seed is multiplied by 37 and all but the fractional portion is discarded. Then to determine a number between the lower boundary and the upper boundary the number is multiplied by the upper and lower and then added to the lower. This gives a real number between the lower and upper. This number is converted to an integer and returned.

i,j,k: Temporary integer loop variables.

MsgPerSec: This is defined as $\text{NumMsg} / \text{NumSes}$

NumPageInst: This is the number of pages in the Instruct.Doc file.

TempBool: This is a temporary boolean.

Strl0: This variable is used for data entry. After the entry is made, Strl0 is checked, if Strl0 has a length of 0 then the old value of the variable is left. If not 0 then Strl0 is converted and the old value is replaced.

Lines: This is the number of lines in the Instruct.Doc file. It is used to calculate the number of pages in the Instruct.Doc.

NumMsg: This is the number of messages to be given to the expert.

TotScales: This is the number of scales or lines found in the file DiffScales.Txt.

Msg,Cat: These are variables used to determine and find the next message to be used in the MsgJudge input file. Cat is the category to be used and Msg is the message in that category to be selected.

Code: This used for the Val procedure. If it comes back 0 then the conversion from string to integer was successful, otherwise an error occurred.

NumCat: This variable will have the number of categories found in the file Messages.Txt.

ActNumMsg: This array will have the actual number of messages from each category found in the file messages.txt.

MsgCount: This variable is used to keep track of how many messages have been pulled from each category. This insures that the same number of messages from each category will be used.

NumSec: This is the number of segments in which the messages are presented to the expert. After each of these segments the expert is given the option of exiting, and saving his data and returning at a later date.

FILE DEFINITIONS:

Messages: This file has all of the original messages. The format of this file is as follows:

```

MsgCat
MessageSpec (i.e. FEE30)
MessageLine1 (each of these MessageLines must be
MessageLine2 at least a carriage return.)
MessageLine3
MessageLine4
MessageLine5
MessageSpec (i.e. FEE30)
MessageLine1 (each of these MessageLines must be
MessageLine2 at least a carriage return.)
MessageLine3
MessageLine4
MessageLine5
..
..
..
..
..
..
..
MsgCat
..
..
..
..
..
..

```

ConFile: This file is the control file for MsgJudge, it supplies MsgJudge with the number of sessions, the number of messages per session, number of pages in the instruction, the seed for the random number generator, the different message environments for the forced question, and the loop control variables:

FixOrder, DiffQuest, and ForceQuest.

The file is of the form:

```
Seed
FixOrder
DiffQuest
ForceQuest
NumPageInst NumSec MsgPerSec TotScales
MsgType1
MsgType2
```

(With each of these lines must be a separate line in the confile and the fifth line each of the variables separated by a space. Note that this file is read in at the beginning of MsgPrep so that the researcher need not enter any of the variables at run time. Any variable that the researcher wishes to leave the same as before, he may enter a carriage return with nothing else on the line. Also note that MsgJudge rewrites the confile after each session which updates the seed.)

Instruct: This file has the instructions and is assigned to the filename 'Instruct.Doc'. The form of this file is simply a text file of 80 character. The researcher should note that 24 lines is displayed at a time. To modify instruct.doc use a word processor in text mode. Do not insert any word wrap or right justification characters. Then rerun MsgPrep. If the same setup is desired, type a carriage return through all questions except for the seed. Enter seed used from last setup. If the exact messages are not required to be displayed again, and only the same number are required, then type a carriage return through all questions.

OutFile: This file specifier is assigned to the filename 'PrepMsg.Txt'. This file will contains the messages that have been randomly chosen by MsgPrep for MsgJudge. These are the messages that will be displayed by MsgJudge to the experts. The format of this file is only required to match the format used in MsgJudge. OutFile format:

```
MessageSpec (i.e. FEE30)
MessageLine1 (each of these MessageLines must be
MessageLine2 at least a carriage return.)
MessageLine3
MessageLine4
MessageLine5
MessageSpec (i.e. FEE30)
MessageLine1 (each of these MessageLines must be
MessageLine2 at least a carriage return.)
MessageLine3
MessageLine4
MessageLine5
```



```

..
..
..
..
..      (this file will have a sixth line
..      at the Beginning of each of the
..      messages)

DiffScales:      This file contains the Differential adjectives to be
                  displayed at each end of the differential scales in msg-
                  judge. Its purpose in MsgPrep is to determine the
                  number of scales being used. To change existing scales,
                  or to add scales or delete scales use a text editor.
                  After completing all changes, re-run MsgPrep.
                  *)

```

(***** Variable Declarations Section *****)

```

Const
  BlankLine =
  '
';

Var
  FixOrder, DiffQuest, SurBool      : Char;
  Ch                                : String[1];
  MsgType1, MsgType2                : String[6];
  ForceQuest                        : String[7];
  Str10                             : String[10];
  Str80                             : String[80];
  Seed                              : Real;
  i, j, k                           : Integer;
  NumSec, MsgPerSec                  : Integer;
  NumPageInst, Lines, NumMsg         : Integer;
  TotScales, Msg, Cat, Code, NumCat  : Integer;
  Messages, ConFile                  : Text;
  Instruct, OutFile, DiffScales      : Text;
  MsgUsed                           : Array [1..10, 1..100] of Boolean;
  MsgCount, ActNumMsg                : Array [1..10] of Integer;
  TempBool                          : Boolean;

```

(*****)

```

Function Ran (Var Seed:Real;Lbound,Ubound:Integer):Integer;

```

```

(* The function produces a random number using a seed passed by the
   main program and updated after each random number is created.
   *)

```

```

  Begin
    Seed := Seed*37.0;
    Seed := Seed - Trunc(Seed);
    Ran  := Trunc((Ubound-Lbound+1)*Seed) + Lbound;
  End;

```

(***** Determine variables from Files *****)

Begin

```
ClrScr;
(* Read in Old Confile *)
Assign (Messages, 'Messages.Txt');
Reset (Messages);
Assign (OutFile, 'PrepMsg.Txt');
Rewrite (OutFile);
Assign (ConFile, 'ConFile.Txt');
Reset (ConFile);
ReadLn (ConFile, Seed);
ReadLn (ConFile, FixOrder);
ReadLn (ConFile, DiffQuest);
ReadLn (ConFile, SurBool);

ReadLn (ConFile, ForceQuest);
ReadLn (ConFile, NumPageInst, NumSec, MsgPerSec, TotScales);
ReadLn (ConFile, MsgType1);
ReadLn (ConFile, MsgType2);
Close (ConFile);
```

(* Find the number of Scales for differentials *)

```
GotoXY (1,1);
Write('Determining number of Differential Scales for MsgJudge. ');
Assign (DiffScales, 'DiffScal.Txt');
Reset (DiffScales);
TotScales := 0;
While Not Eof(DiffScales) do
  Begin
    TotScales := TotScales + 1;
    ReadLn (DiffScales);
  End;
Close (DiffScales);
```

(* Find the number of Pages in Instructions *)

```
GotoXY (1,2);
Write('Determining number of Pages in instruction for MsgJudge. ');
Assign (Instruct, 'Instruct.Doc');
Reset (Instruct);
Lines := 0;
While Not Eof(Instruct) do
  Begin
    Lines := Lines + 1;
    ReadLn (Instruct);
  End;
NumPageInst = Lines Div 24;
If Lines Mod 24 <> 0 then NumPageInst := NumPageInst + 1;
Close (Instruct);
```

(* Find the number of messages in the Message File *)

```

GotoXY (1,3); Write ('Determining number of Messages and Categories.');
```

NumCat := 0;

For i := 1 to 10 do ActNumMsg[i] := 0;

Repeat

```

    ReadLn (Messages, Str80);
    If Str80 = 'MsgCat' then
        Begin
            ReadLn(Messages);
            NumCat := NumCat + 1;
        End;
    For i := 1 to 5 do ReadLn (Messages);
    ActNumMsg[NumCat] := ActNumMsg[NumCat] + 1;
Until Eof (Messages);

(***** Prompted for other needed Variables *****)

(* Find number of message to be used and the number of Sessions *)
NumMsg := MsgPerSec * NumSec;
Code := 0;
Repeat
    GotoXY (1,5);
    Write
    ('Enter the total number of message to be displayed to expert ?');
    ReadLn (Str10);
    If Str10 <> '' then Val (Str10, NumMsg, Code); (* If not empty string
    *)

    If Code <> 0 then (* If Bad integer then *)
        Begin
            GotoXY (1,6);
            Write('Error, not a legal integer, strike any key to
continue...');
            While Not(KeyPressed) do;
                GotoXY (1,5); Write (BlankLine);
                GotoXY (1,6); Write (BlankLine);
            End;
            (* If there can not be the same number of message for each Cat *)
            If NumMsg Mod NumCat <> 0 then
                Begin
                    GotoXY (1,6);
                    Write('Error, number of messages must be divisible by
                    ', NumCat);
                    GotoXY (1,7); Write ('Strike any key to continue...');
                    While Not(KeyPressed) do;
                        GotoXY (1,5); Write (BlankLine);
                        GotoXY (1,6); Write (BlankLine);
                        GotoXY (1,7); Write (BlankLine);
                    Code := 1;
                End;
        End;

```

```

Until Code = 0;  (* Until good integer *)
Repeat
  GotoXY (1,6);
  Write
  ('Enter the number of sessions for the messages to be divided
  into?');
  Readln (Str10);
  If Str10 <> '' then Val (Str10,NumSec,Code); (* If not empty string
  *)

  If Code <> 0 then (* Bad integer then *)
    Begin
      GotoXY (1,7);
      Write('Error, not a legal integer, strike any key to
  continue...');
      While Not(KeyPressed) do;
        GotoXY (1,6); Write (BlankLine);
        GotoXY (1,7); Write (BlankLine);
      End

    Else (* if legal test range *)
      Begin
        If NumSec <= 0 then (* Check if less than or equal to zero *)
          Begin
            GotoXY (1,7);
            Write ('Error, number of section must at least 1, strike any
  key to continue...');
            While Not(KeyPressed) do;
              GotoXY (1,6); Write (BlankLine);
              GotoXY (1,7); Write (BlankLine);
            End;
            (* Check if greater than Number of messages *)
            If NumSec > NumMsg then
              Begin
                GotoXY (1,7);
                Write ('Error, number of section must at less then ',NumMsg);
                GotoXY (1,8); Write ('Strike any key to continue...');
                While Not(KeyPressed) do;
                  GotoXY (1,6); Write (BlankLine);
                  GotoXY (1,7); Write (BlankLine);
                  GotoXY (1,8); Write (BlankLine);
                End;
              End;
            End; (* End of Check range *)

          MsgPerSec := NumMsg Div NumSec;
          (* Check to see if NumMsg can be divided into Numsec sections *)
          If NumMsg Mod NumSec > 1 then

```

```

Begin
  GotoXY (1,7);
  Write
    ('Error, Number of Messages not divisible by number of sections.');
```

GotoXY (1,8); Write ('Strike any key to continue...');

```

  While Not(KeyPressed) do;
    GotoXY (1,6); Write (BlankLine);
    GotoXY (1,7); Write (BlankLine);
    GotoXY (1,8); Write (BlankLine);
  Code := 1;
End;
Until Code = 0; (* Until good integer *)

(* Find Seed to be used for random generator *)
Repeat
  GotoXY(1,7); Write (BlankLine);
  GotoXY(1,7); Write ('Enter Seed to be used for randomization ? ');
  Read (Str10);
  If Str10 <> '' then Val (Str10,Seed,Code); (* If not empty string *)
  If Code <> 0 then (* If Bad real then *)
    Begin
      Write('Error, not a legal real, strike any key to continue...');
      While Not(KeyPressed) do;
        GotoXY (1,7); Write (BlankLine);
        GotoXY (1,8); Write (BlankLine);
      End
    End
  Else (* Else see if it is in range *)
    If ((Seed <= 0) or (Seed >= 1)) then
      Begin
        GotoXY (1,8);
        Write
          ('Error, Seed must be greater than 0 and less than 1.');
```

GotoXY (1,9); Write ('Strike any key to continue...');

```

        While Not(KeyPressed) do;
          GotoXY (1,7); Write (BlankLine);
          GotoXY (1,8); Write (BlankLine);
          GotoXY (1,9); Write (BlankLine);
          Code := 1; (* Set code to bad integer *)
        End;
      Until (Code = 0); (* Until good integer *)

(* Find Out rest of Info for ConFile *)
Repeat
  GotoXY (1,9);
  Write ('Do you wish to give messages in the same random order to');
  GotoXY (1,10); Write ('each expert ? (Y/N) ');
  Read (Str10);
  If Str10 <> '' then (* If not empty string *)
    Begin
      FixOrder := UpCase (Str10[1]);
      (* Make sure a legal answer, a Y or N *)
      If Not ((FixOrder = 'Y') or (FixOrder = 'N')) then
```

```

        Begin
            GotoXY (1,11);
Write ('Please enter Y for yes or N for No. ');
            GotoXY (1,12); Write ('Strike any key to continue...');
            While Not(KeyPressed) do;
                GotoXY (1,10); Write (BlankLine);
                GotoXY (1,11); Write (BlankLine);
                GotoXY (1,12); Write (BlankLine);
            End;
        End;
    Until ((FixOrder = 'Y') or (FixOrder = 'N'));
Repeat
    GotoXY (1,11);
    Write ('Do you wish to give differential questions ? ');
    Read (Strl0);
    If Strl0 <> '' then (* If not empty string *)
        Begin
            DiffQuest := UpCase (Strl0[1]);
            If Not ((DiffQuest = 'Y') or (DiffQuest = 'N')) then
                Begin
                    GotoXY (1,12);
Write ('Please enter Y for yes or N for No. ');
                    GotoXY (1,13); Write ('Strike any key to continue...');
                    While Not(KeyPressed) do;
                        GotoXY (1,11); Write (BlankLine);
                        GotoXY (1,12); Write (BlankLine);
                        GotoXY (1,13); Write (BlankLine);
                    End;
                End;
            End;
            If DiffQuest = 'N' then ForceQuest := 'After';
        Until ((DiffQuest = 'Y') or (DiffQuest = 'N'));
        If DiffQuest = 'Y' then
            Repeat
                (* Display options (a) (b) (n) *)
                GotoXY (1,13);
                Write ('When do you wish to administer choice questions : ');
                GotoXY (5,15);
                Write ('(A)fter all differential questions for all messages have
                    been asked. ');
                GotoXY (5,16);
                Write ('(B)efore the differential questions for each message. ');
                GotoXY (5,17); Write ('(N)o message type question at all. ');
                GotoXY (1,18);
                Write ('Enter first letter ? ');
                GotoXY (60,18); ReadLn (Ch);
                If Ch = '' then Ch := ' ';
                TempBool := False;
                Case Ch of
                    'A','a' : Begin
                        ForceQuest := 'After';
                        TempBool := True;
                    End;

```

```

    'B','b' : Begin
        ForceQuest := 'Before';
        TempBool := True;
    End;
    'N','n' : Begin
        ForceQuest := 'None';
        TempBool := True;
    End;
    ' ' : Begin
        TempBool := True;
    End;
End; (* End of Case Statement *)
If TempBool = False then
    Begin
        GotoXY (1,20);
        Write ('Illegal character, strike any key to continue...');
        While Not(KeyPressed) do;
            GotoXY (1,19); Write (BlankLine);
            GotoXY (1,20); Write (BlankLine);
        End;
    Until TempBool = True;
(* End of If DiffQuest = 'Y' *)

Repeat
    GotoXY (1,19);
    Write ('Do you wish to give a survey ? ');
    Read (Str10);
    If Str10 <> '' then (* If not empty string *)

        Begin
            SurBool := UpCase (Str10[1]);
            If Not ((SurBool = 'Y') or (SurBool = 'N')) then
                Begin
                    GotoXY (1,20);
Write ('Please enter Y for yes or N for No. ');
                    GotoXY (1,21); Write ('Strike any key to continue...');
                    While Not(KeyPressed) do;
                        GotoXY (1,19); Write (BlankLine);
                        GotoXY (1,20); Write (BlankLine);
                        GotoXY (1,21); Write (BlankLine);
                    End;
                End;
            Until ((SurBool = 'Y') or (SurBool = 'N'));

            GotoXY (1,21);
            Write ('Enter abr. for first set of messages being compared ? (Max 6
                letters) ');
            ReadLn (Str10);
            (* If not empty string *)
            If Str10 <> '' then For i := 1 to Length(Str10) do
                MsgType1[i] := UpCase(Str10[i]);
            GotoXY (1,22);
            Write ('Enter abr. for second set of messages being compared ? (Max 6
                letters) ');

```

```

ReadLn (Str10);
(* If not empty string *)
If Str10 <> '' then For i := 1 to Length(Str10) do
    MsgType2[i] := UpCase(Str10[i]);

GotoXY (1,24); Write ('Creating input files for MsgJudge...');
(* Write out new confile *)
Rewrite (ConFile);
WriteLn (ConFile,Seed);
WriteLn (ConFile,FixOrder);
WriteLn (ConFile,DiffQuest);
WriteLn (ConFile,SurBool);
WriteLn (ConFile,ForceQuest);
WriteLn (ConFile,NumPageInst:3,NumSec:3,MsgPerSec:3,TotScales:3);
WriteLn (ConFile,MsgType1);
WriteLn (ConFile,MsgType2);
(* Set all message to non-used *)
For i := 1 to 4 do
    Begin
        MsgCount[i] := 0;
        For j := 1 to 100 do MsgUsed[i,j] := False;
    End;

(* Find NumMsg worth of messages and place each of them in prepmsg file *)
For i := 1 to NumMsg do
    Begin
        (* Find starting point in Categories *)
        Cat := Ran(Seed,1,NumCat);
        (* Search for available Category *)

        While MsgCount[Cat] >= MsgPerSec do
            Begin
                Cat := Cat + 1;
                If Cat > NumMsg Div MsgPerSec then Cat := 1;
            End;

        (* Find starting point in message for Cat category *)
        Msg := Ran(Seed,1,ActNumMsg[Cat]);
        (* Search for next available msg *)
        While MsgUsed[Cat,Msg] do
            Begin
                Msg := Msg + 1;
                If Msg > ActNumMsg[Cat] then Msg := 1;
            End;
        MsgUsed[Cat,Msg] := True; (* Set message to used *)
        MsgCount[Cat] := MsgCount[Cat] + 1; (* Increase count *)
        Reset (Messages);
        For j := 1 to Cat do (* Search for Category to be used *)

```



```

        Repeat
            Readln (Messages,Str10);
            Until (Str10 = 'MsgCat');
        For j := 1 to Msg-1 do (* Search for message in category *)
            For k := 1 to 6 do ReadLn (Messages);
            Readln (Messages,Str10); (* Read Message Specifier *)
            Writeln (Outfile,Str10); (* Write Message Specifier *)
            For j := 1 to 5 do
                Begin
                    Readln (Messages,Str80); (* Read a line of message *)
                    Writeln (OutFile,Str80); (* Write a line of message *)
                End;
            MsgUsed[Cat,Msg] := True;
        End;
    (* Close output files *)
    Close (OutFile);
    Close (ConFile);
End.

```

APPENDIX B

MSGJUDGE PROGRAM

Below is the source code listing for the MSGJUDGE program. It is written in TurboPascal and runs on an IBM PC.

Program MsgJudge (Input,Output);

(*

The purpose of this program is to determine the differences between messages that originate in two different environments. The differences are determined by asking force-choice questions of the expert judge (i.e. From which environment do you think this message was taken), and secondly by presenting the message with a differential adjective set, and asking the judge to rate the subject message based on the adjective sets. The differential adjectives are placed on a scale from -3.5 to 3.5. A pointer is used to determine the value the expert wishes to enter. Due to the length of time required to rate all of the messages, the expert is given the option of leaving the program at pre-determined intervals and returning at a later time. If this choice is made, the data collected is saved and the expert will restart the program at the same place that it was exited. These intervals are pre-determined and set while running the MsgPrep program.

After all differential questions have been asked, and after the force-choice questions, a survey is administered. The questions for the survey are found in the file 'Survey.Txt'. For more information on the form of this survey file, refer to the file definition section.

MsgJudge is setup by the program MsgPrep. MsgPrep sets up the Confile and PrepMsg files for MsgJudge. There are two other files that also must be present: Instruct.Doc (displays instructions to the expert when first entering MsgJudge) and DiffScal.Txt (which contains the differential adjective sets).

Input files required for MsgJudge:

- | | |
|------------------|------------------------------------|
| (1) Instruct.Doc | (Written by researcher) |
| (2) ConFile.Txt | (Setup by MsgPrep) |
| (3) PrepMsg.Txt | (Created by MsgPrep) |
| (4) DiffScal.Txt | (Created by researcher) |
| (5) Survey.Txt | (Created by researcher "Optional") |

Survey.Txt is only required if the researcher has selected that a survey be administered.

BASIC PROGRAM FLOW:

- (1) Output filename construction
 - i. Ask for Initials
 - ii. Ask for month
 - iii. Ask for day
 - iv. Build filename prefix (Initials+Month+Day)

- (2) Initialize variables, initialize files and give instructions.
- (3) Give questions for each message.
 - i. Determine next message.
 - ii. If ForceQeust = 'Before', give forced question.
 - iii. Give differential scale and get response.
 - a. Determine next scale.
 - b. Determine orientation.
 - c. Display scale.
 - d. Get response.
 - e. Translate response based on orientation of scale.
- (4) If ForceQuest = 'After' and all differential scales are complete:
 - i. For each message give forced question.
- (5) If all differential scales are done, conduct survey.
- (6) Write Outfiles.
- (7) Close output files.

VARIABLES DEFINITIONS:

Continue: Loop control for continue option at the end of each session.

FixOrder: This is a character that defines the order in which msgjudge presents the messages. If yes, the messages are given to all experts in the same random order found in the input file PrepMsg.Txt. If no, then the messages are randomized again by MsgJudge, and the experts will see the messages displayed in a different random order.

SurBool: This boolean will turn the Survey off and on in MsgJudge.

DiffQuest: This character is used to turn on and off the differential question. If yes then the expert is given each message with the differential adjective scales. If no then the expert gets no differential scales and the forcequest variable is automatically defined as after.

Ch: Temporary Character variable.

Month: String of length 2, which will store the month of the year. This value is used to create the names of the output files.

Day: String of length 2, stores the day of the month and is used to create the output filenames.

Initials: This variable is also used to create the output filename. It contains the experts initials and is a String length of 2.

MsgType1: This variable is used for the force-choice question. It defines the first environment for the force-choice question.

MsgType2: This variable is also used for the force-choice question. It defines the second environment.

Dummy: Used for input dummy variable. It is used to advance file pointer forward six characters.

ForceQuest: This variable defines when, if at all, the force-choice question will be presented to the expert. The following are the choices for ForceQuest:

- (1) 'After' - this option will present the forced question after all the differential scales have been answered for all the messages.
- (2) 'Before' - this option will present the forced question before the differential scales for each of the messages. In other word, for each message the forced question is asked first then the differential scales are asked.
- (3) 'No'- no forced question to be asked.

Filename: Filename of output file. This file contains all the answers made by the expert to the differential questions, the forced choice question, and the survey answers. The filename is Initials+Month+Dat+'.Dat'.

Stat: This is a filename used for the Status file. This file contains all information needed for re-starting the program and returning to same point in execution. This file is created every time the expert is asked if he wishes to continue and he responds no. This file is erased from the directory as soon as the full execution of all phases of the program is complete.

SurTyp: Is used in the survey section. It will be read from the file 'Survey.txt' and will have one of three values. 'Reset' which will clear the screen and reset question position variable, 'Prompt' which takes the next three lines and writes them on the first three lines of the screen, or 'Question' which will take the next three lines of the file and places all non blank lines to the screen.

Str80: This variable is used for reading a line of 80 characters from a file for display purposes. It is used in the survey section to display text and for writing messages.

TypAns: This array is used to store the answers to the forced questions. There is a position for each of the messages.

DiffAns: This array is used to store the answers to the differential scales for each of the messages.

Msg: Is an array for storing each of the messages and each of its five lines.
..

MsgSpe: Is an array which stores the message specifier. These are used in the output so that the answers can be related to the original messages.

MsgUsed: Is an array used to keep track of which messages have already been used. This is only important in the mode where the messages are not taken in the order that they are found in the file 'PrepMsg.Txt'. It should be noted that messages in 'PrepMsg.Txt' are in random order. The random order is used when you wish the message to be given in a different random order to each of the experts.

ScaleUsed: This is used to keep track of those scales that have already been used. This option unlike the MsgUsed is always used.

Diff: Array where the differential scale adjectives are stored.

SurAns: Array where the answers to the survey questions are stored.

Seed: This variable is a real number greater than 0 and less than 1. It is used to determine the next random number. First, the seed is multiplied by 37 and all but the fractional portion is discarded. Then to determine a number between the lower boundary and the upper boundary the number is multiplied by the upper and lower and then added to the lower. This gives a real number between the lower and upper. This number is converted to an integer and returned.

i,j,k: Temporary integer loop variables.

StPtr: Is used to advance the messages to the MsgJudge section being administered.

NumSec: This is the number of segments that the messages are presented to the expert. After each of these segments the expert is given the option of exiting, saving his data and returning at a later date.

Quest: This variable is used to advance through the SurAns array. It is then used to determine how many answers need to be written out to the OutFile.

LineLen: Integer used to store the length of a line read in. It is used in writing that line to the screen.

Section: Holds the value of the current section being administered.

MsgPerSec: This is defined as NumMsg / NumSes.

TotScales: Number of scales; read from ConFile.

MsgCount: Number of messages already administered in the differential question portion of the program.

ScaleCount: Number of scales already administered in the differential question portion of the program.

MsgNum: The message number that is chosen for display.

Scale: The scale number that is chosen for display.

Orientation: The random orientation of the + and - scales for the differential scales.

PointerPos: Random end of the scales for the pointer to begin at. It is then used in GetDispAns for position of pointer on Scale.

NumPageInst: This is the number of pages in the Instruct.Doc file.

TempBool: This is a temporary boolean.

FILE DEFINITIONS:

ConFile: This file is the control file for MsgJudge, it supplies MsgJudge with the number of Sessions, the number of messages per session, number of pages in the instructions, the Seed for the random number generator, the different message environment for the forced questions, and the loop control variables: FixOrder, DiffQuest, and ForceQuest.

The file is of the form:

Seed

FixOrder

DiffQuest

ForceQuest

NumPageInst NumSec MsgPerSec TotScales

MsgType1

MsgType2

With each of these lines being on a separate line in the confile and the fifth line of each of the variables separated by a space.

Instruct: This file has the instructions and is assigned to the filename 'Instruct.Doc'. The form of this file is simply a text file of 80 character. The researcher should note that 24 lines is displayed at a time. To modify instruct.doc use a word processor in text mode. Do not insert any word wrap or right justification characters. Then rerun MsgPrep. If the same setup is desired, type a carriage return through all questions except for the seed. Enter seed used from last setup. If the exact messages are not required to be displayed again, and only the same number are required, then type a carriage return through all questions.

Survey: A text file of the name 'Survey.Txt'. This file is created manually with a word processor using a Non-Documen mode. The utility used here allows for three modes, and are as follows :

- (1) 'Reset', if this appears in the survey file then the screen is cleared and the line marker is moved to the top of question section of the screen. No blank lines may appear after a reset. The next line is expected to be a command.
- (2) 'Prompt', this feature allows researcher to prompt expert on the intentions of the following questions and how he/she should answer the question. Basically this feature will take the next three lines from the survey file and print them on the top three lines of the screen. With the prompt there will always be three lines after the prompt command, and the fourth is expected to be a command.
- (3) 'Question', this command will take the next three lines from the survey file and will print all non blank lines to the screen. Also the line counter is advanced for all non blank lines. This is so that no lines are wasted on the display and more question may be displayed at the same time. With the question command there will always be three lines after the question command. They may be blank. The fourth line after the question command, will be a command. These three command are the only things that can be on the first line of the survey file and the rules about the number of lines that must appear after each command are the only rules for this file. After the required number of lines for each command, one of two thing must occur. There will be a command on the next line or End of File will be found.

Infile: This file specifier is assigned to the filename 'PrepMsg.Txt'. This file will contain the messages that have been randomly chosen by MsgPrep for MsgJudge. These are the messages that will be displayed by MsgJudge to the expert. The format is as follows:

```

MessageSpec (i.e. FEE30)

MessageLine1 (each of these MessageLines must be at
MessageLine2 a carriage return.)
MessageLine3
MessageLine4
MessageLine5

MessageSpec (i.e. FEE30)

MessageLine1 (each of these MessageLines must be at
MessageLine2 a carriage return.)
MessageLine3
MessageLine4
MessageLine5

```

..

..

..

..

.. (this file will have six lines at ..
the Beginning for each of
the NumMsg.)

DiffScales: This file contains the differential adjectives to be written at each end of the differential scales that are displayed in msgjudge. To change the number of Scales, add or delete the Scales that are desired then rerun msgprep.

OutFile: The outfile has all the answers from the expert to the different parts of MsgJudge. It contains the answers to the differentials, the answers to the forced questions and finally the answers to the survey questions. The outfile is named by the concatenation of Initials+Month+ Day+'.Dat'. The outfile is of the form:

Scale	1	2	3	4	5	6	7
-------	---	---	---	---	---	---	---

MsgSpe DiffAns

..

..

.. (Differential answers, one for each message)

.. MsgSpe TypAns

..

.. (Forced question answers, one for each message)

..

1 Answer first question

2 Answer second question

.. (Survey question answers, one for each

.. question. Number of question determined

.. by number of question found in 'Survey.Txt')

*)

(***** Actual Variable Declarations *****)

Const

BlankLine =

,

;

Type

Ch4 = String[4];

Str12 = String[12];

TypChar = Array [1..120] of Char;

DiffArr = Array [1..120,1..7] of Real;

MsgArr = Array [1..120,1..5,1..80] of Char;

MsgSArr = Array [1..120] of String[5];

DiffScArr = Array [1..10,0..1] of String[12];

SurArr = Array [1..120] of String[10];

Var

Continue,FixOrder,DiffQuest : Char;
SurBool,SurDone,ForceDone,Ch : Char;
Month,Day : String[2];
Initials : String[3];
MsgType1,MsgType2,Dummy : String[6];
ForceQuest : String[7];
Filename,Stat,SurTyp : String[12];
Str80 : Array [1..80] of Char;
TypAns : TypChar;
DiffAns : DiffArr;
Msg : MsgArr;
MsgSpe : MsgSArr;
MsgUsed : Array [1..120] of Char;
ScaleUsed : Array [1..7] of Boolean;
Diff : DiffScArr;
SurAns : SurArr;
QuestType : String[6];
QuestChar : Char;

ConFile,OutFile,Infile,Status,DiffScales,Survey : Text;
Bool,TempBool : Boolean;
Seed : Real;
XCursor,YCursor : Integer;
i,j,k,Temp,StPtr,NumSec,Quest,LineLen : Integer;
Section,MsgPerSec,TotScales,MsgCount,ScaleCount : Integer;
MsgNum,Scale,Orientation,PointerPos,NumPageInst : Integer;

```

(***** Start of Utilities *****)

Function Convert (Str:Ch4;Var IntVal:Integer) : Boolean;

(* This function will convert a string into an integer and return the
   value in IntVal. The function itself will return True or False.
   True if conversion was successful, false otherwise.
*)

Var
    i,Result : integer;
    TempBool : Boolean;

Begin
    Val (Str,IntVal,Result);
    If Result = 0 then
        Convert := True
    Else
        Convert := False;
End; (* End of Procedure Convert *)

Function Ran (Var Seed:Real;Lbound,Ubound:Integer):Integer;

(* Function produces random number using a seed passed by main
   program and updated after each random number is created.
*)

Begin
    Seed := Seed*37.0;
    Seed := Seed - Trunc(Seed);
    Ran := Trunc((Ubound-Lbound+1)*Seed) + Lbound;
End;

```

Procedure ViewText (Filename:Str12;Page,MaxPage:Integer);

(* ViewText will allow researchers to print any page of a text file to the screen. Then allow them to move a page previous or a next page. When researcher wishes to return they type a 'C'.

*)

Var

i : Integer;

Ch : Char;

Procedure ViewPage (Filename:Str12;Page:Integer);

(* This procedure will print the desired page for ViewText *)

Var

i,j : Integer;

TxtFile : Text;

Begin

Assign (TxtFile,Filename);

Reset (TxtFile);

(* Find desired page *)

For i := 1 to Page-1 do

For j := 1 to 24 do

If Not(Eof(TxtFile)) then Readln (TxtFile);

ClrScr;

(* Write desired page to screen *)

```

For i := 1 to 24 do
    If Not Eof(TxtFile) then
        Begin
            GotoXY (1,i);
            While Not EoLn(TxtFile) do
                Begin
                    Read(TxtFile,Ch);
                    Write(Ch);
                End;
            If Not Eof(TxtFile) then Readln (TxtFile);
        End;
    Close(TxtFile);
End;
Begin
    Repeat
        ViewPage(Filename,Page);
        GotoXY(1,25); Write(BlankLine);
        GotoXY(1,25);
        Write('(C)ontinue program execution, (N)ext Page');
        Write(', (P)revious Page');
        Read (Kbd,Ch);
        Case Ch of
            'N','n':Begin
                Page:=Page+1;
                If Page > MaxPage then Page := MaxPage;
            End;

```

```

        'P','p':Begin
            Page:=Page-1;
            End;
        End; (* End of Case *)
    Until ((Ch='C')or(Ch='c'));
End;

(***** Initialize variable for first time *****)

Procedure Init;

    Var
        j,k : Integer;
    Begin
        For j := 1 to 120 do            (* For each message *)
            Begin
                TypAns[j] := ' ';
                For k := 1 to 7 do      (* For each Scale *)
                    Begin
                        DiffAns[j,k] := 5;
                    End;
                MsgUsed[j] := 'F';
            End;
        End;
    End;

```

(***** Read and Write File Procedures *****)

Procedure ReadInfiles;

Var

i,j,k,l : Integer;

Begin

Assign (Infile,'PrepMsg.txt');

Reset (Infile);

Assign (DiffScales,'DiffScal.Txt');

Reset (DiffScales);

(* Read PrepMsg file *)

For i := 1 to 120 do

Begin

(* Read Message specifier *)

Readln (InFile,MsgSpe[i]);

.(*) Read in a message *)

For j := 1 to 5 do

Begin

For k := 1 to 80 do

Begin

If Not Eof(Infile) then

If (Not Eoln(Infile)) then

Read (Infile,Msg[i,j,k])

Else

Msg[i,j,k] := ' '

Else

Msg[i,j,k] := ' ';

End;

```

        Readln (Infile);

    End;

End;
    ..

(* Read in Differentials Adjectives *)
For i := 1 to TotScales do
    Begin
        Readln (DiffScales,Diff[i,0],Diff[i,1]);

    End;

    Close (Infile);

    Close (DiffScales);

End;

Procedure ReadOutFiles;

Var

    i,j,k : integer;

Begin

    Reset (OutFile);

    (* Read in previously answered messages *)

    Readln (OutFile);

    ReadLn (OutFile);

    (* Read in answer for differentials scales *)

    For i := 1 to NumSec * MsgPerSec do

        Begin

            Read (OutFile,Dummy);

            For j := 1 to 7 do

                Read (OutFile,DiffAns[i,j]);

            Readln (OutFile);

        End;

    End;

```



```

    Readln (OutFile);

    (* Read in answers to forced question *)

    For i := 1 to NumSec * MsgPerSec do
        Read (OutFile,Dummy,TypAns[i]);

    End;

Procedure WriteOutFile;

Var

    i,j,k : Integer;

Begin

    Rewrite (OutFile);

    Writeln (OutFile,'Scale    1    2    3    4    5    6    7');
    Writeln (OutFile,'-----');

    (* Write out differential scales answers *)

    For i := 1 to NumSec * MsgPerSec do

        Begin

            (* Write out message specifier *)

            Write (OutFile,MsgSpe[i],' ');

            (* Write out differential scales ans *)

            For j := 1 to 7 do

                Write (OutFile,DiffAns[i,j]:5:1);

            Writeln (OutFile);

        End;

    Writeln (OutFile);

    (* Write out forced question answers *)

    For i := 1 to NumSec * MsgPerSec do

        Writeln (OutFile,MsgSpe[i],' ',TypAns[i]);

    Writeln (OutFile);

```

```

    (* Write out answers to survey *)

    For i := 1 to Quest do

        Writeln (OutFile,i:2,' ',SurAns[i]);

    Close (OutFile);

End;

(***** Survey Read and Write procedures *****)

Procedure ReadLine;

Var

    i : Integer;

Begin

    (* Read a line from survey *)

    LineLen := 1;

    While ((Not Eoln(Survey)) and (LineLen <= 80)) do

        Begin

            Read (Survey,Str80[LineLen]);

            LineLen := LineLen + 1;

        End;

    Readln (Survey);

    For j := LineLen to 80 do Str80[j] := ' ';

End;

Procedure WriteLine;

(* display line *)

Var

    i : integer;

```

Begin

 If LineLen > 80 then LineLen := 80;

 For i := 1 to LineLen do Write (Str80[i]);

 ..

End;

(***** Get Answer Procedures *****)

Function GetAnsDisp (Scale,Orientation,PointerPos:Integer):Real;

(* This function will display the scale to answered and allow expert
to move a pointer to location desired on scale. When the desired
location has been found the position is translated into the
appropriate value according to the orientation.

*)

Var

 Ch : char;

 OldPointerPos : Integer;

Begin

 GotoXY (1,15); Write(BlankLine);

 GotoXY (1,16); Write(BlankLine);

 GotoXY (1,11); Write(BlankLine);

 GotoXY (2,14); Write

(" ----+----| ----+----| ----+----| ----+----| ----+----| ----+----| ');

Case Orientation of

0 : Begin

GotoXY (4,11); Write (Diff[Scale,0]);

GotoXY (63,11); Write (Diff[Scale,1]);

GotoXY (2,13); Write

(' 3 2 1 0 -1 -2 -3');

End;

1 : Begin

GotoXY (4,11); Write (Diff[Scale,1]);

GotoXY (63,11); Write (Diff[Scale,0]);

GotoXY (2,13); Write

(' -3 -2 -1 0 1 2 3');

End;

End; (* End of Case Orientation *)

If PointerPos = 1 then

Begin

PointerPos := 71;

OldPointerPos := 1;

End

Else

Begin

PointerPos := 1;

OldPointerPos := 71;

End;

Repeat

GotoXY (OldPointerPos+1,15); Write(' ');

GotoXY (OldPointerPos+1,16); Write(' ');

GotoXY (PointerPos+1,15); Write ('#');

GotoXY (PointerPos+1,16); Write ('| ');

GotoXY (1,25);

OldPointerPos := PointerPos;

Read (Kbd,Ch);

Case Ch of

#60 : Begin

PointerPos := PointerPos - 5;

If PointerPos < 1 then PointerPos := 1;

End;

#62 : Begin

PointerPos := PointerPos + 5;

If PointerPos > 71 then PointerPos := 71;

End;

#44 : Begin

PointerPos := PointerPos - 1;

If PointerPos < 1 then PointerPos := 1;

End;

#46 : Begin

PointerPos := PointerPos + 1;

If PointerPos > 71 then PointerPos := 71;

End;

End; (* End Case for Arrow Keys *)

```

Until (Ch = #13);

Case Orientation of

    1 : GetAnsDisp := (PointerPos - 36)/10;
      ..
    0 : GetAnsDisp := (36 - PointerPos)/10 ;

End; (* End of Case *)

End; (* End of Procedure GetAnsDisp *)

Procedure GetTypAns(MsgNum: Integer);

(* This procure display and get the answer to the forced question. The
   message needs to already be displayed.

*)

Begin
    Repeat
        GotoXY (1,9); Write (BlankLine);
        GotoXY (1,10); Write (BlankLine);
        GotoXY (1,9);
        Write('Is this Message from the ',MsgType1,' or ',MsgType2,'
?');

        GotoXY (1,10); Write ('Enter first character from choices : ');
        Read (TypAns[MsgNum]);
        TypAns[MsgNum] := UpCase(TypAns[MsgNum]);
        GotoXY (1,9); Write (BlankLine);
        GotoXY (1,10); Write (BlankLine);
        If Not((TypAns[MsgNum]=MsgType1[1])or
(TypAns[MsgNum]=MsgType2[1])) then

```

```

        Begin
            GotoXY (1,9);
            Write ('Illegal entry strike any key to continue...');
            While Not(KeyPressed) do;
                End;
            Until ((TypAns[MsgNum]=MsgType1[1])or(TypAns[MsgNum]=MsgType2[1]));
        End;

```

(*****)

```

Begin
    ClrScr;
    GotoXY (1,1); Write ('Input, Output File Name Construction :');
    GotoXY (1,3); Write ('Enter your initials ? ');
    Read (Initials);
    Repeat
        GotoXY (1,5); Write ('Enter Month "01-12" ? ');
        Read (Month);
        If Length(Month) = 1 then Month := ConCat ('0',Month);
        TempBool := Convert (Month,Temp);

        If TempBool = False then
            Begin
                GotoXY (1,5); Write (BlankLine);
                GotoXY (40,5); Write ('Illegal integer ...');
            End
        Else
            If ((Temp < 1) or (Temp > 12)) then

```

```

    Begin
        GotoXY (1,5); Write (BlankLine);
        GotoXY (40,5); Write ('Value out of range ...');
    End;
Until ((TempBool = True) and ((Temp >= 1) and (Temp <= 12)));
Repeat
    GotoXY (1,6); Write ('Enter Day of Month "01-31 ? ');
    Read (Day);
    If Length(Day) = 1 then Day := ConCat ('0',Day);
    TempBool := Convert (Day,Temp);
    If TempBool = False then
        Begin
            GotoXY (1,6); Write (BlankLine);
            GotoXY (40,6); Write ('Illegal integer ...');
        End
    Else
        If ((Temp < 1) or (Temp > 31)) then
            Begin
                GotoXY (1,6); Write (BlankLine);
                GotoXY (40,6); Write ('Value out of range ...');
            End;
Until ((TempBool = True) and ((Temp >= 1) and (Temp <= 31)));

Assign (ConFile,'ConFile.txt');
Reset (ConFile);
Readln (ConFile,Seed);
Readln (ConFile,FixOrder);
Readln (ConFile,DiffQuest);

```



```

ReadLn (ConFile,SurBool);

ReadLn (ConFile,ForceQuest);

ReadLn (ConFile,NumPageInst,NumSec,MsgPerSec,TotScales);

ReadLn (ConFile,MsgType1);

ReadLn (ConFile,MsgType2);


GotoXY (1,8); Write ('File Name Construction Complete...');

FileName := ConCat (Initials,Month,Day,'.Dat');

Assign (OutFile,Filename);

$I- Reset(OutFile); $I+

Bool := (IOresult = 0);


Stat := ConCat (Initials,Month,Day,'.Inf');

Assign (Status,Stat);

(* Depending of where continuation or not initialize variables *)

If Bool then

    Begin

        (* If .Inf file exist then reset section *)

        $I- Reset(Status); $I+

        Bool := (IOresult = 0);

        If Bool then

            Begin

                ReadLn (Status,Section);

                ReadLn (Status,ForceDone);

                ReadLn (Status,SurDone);

                For i := 1 to MsgPerSec * NumSec do

                    ReadLn (Status,MsgUsed[i]);

                Close (Status);

            End

        End

    End

```

```

Else
    Begin
        GotoXY (1,14);
        Write ('Continuation mode assumed, but No .Inf file
found...');
        GotoXY (1,15);
        Write ('MsgJudge execution halted.');
```

Halt;
End;
GotoXY (1,14);
Write ('Input Files present, continuation mode assumed...');
GotoXY (1,15); Write ('Strike any key to continue...');
While Not(KeyPressed) do;
ReadOutFiles;
Close (OutFile);
Assign (OutFile,Filename);
End
Else
 Begin
 GotoXY (1,14);
 Write ('Input Files not found creation mode assumed...');
 GotoXY (1,15); Write ('Strike any key to continue...');
 While Not(KeyPressed) do;
 Init;
 Section := 0;
 ForceDone := 'N';
 SurDone := 'N';

```

    For i := 1 to 120 do MsgUsed[i] := 'F';
    ViewText ('Instruct.Doc',1,NumPageInst);
    ClrScr;
End;

ClrScr;
GotoXY (1,1); Write ('Please wait reading input files...');
ReadInfiles;
ClrScr;

For i := 1 to 120 do
    SurAns [i] := '';

(* Start of Main Looping *)
TempBool := True;
Ch := 'Y';
If ((DiffQuest = 'Y') and (Section < NumSec)) then
Begin
    Repeat (* Until all 4 section or continue not true *)
        Section := Section + 1;
        MsgCount := 0;
        Repeat
            MsgCount := MsgCount + 1;
            For i := 1 to 7 do
                ScaleUsed[i] := False;

            (* Find Next message *)
            StPtr := Trunc ((Section-1) * MsgPerSec);
            If Not (FixOrder = 'Y') then

```

```

Begin
    MsgNum := Ran(Seed,1,MsgPerSec*NumSec);
    (* Search for next available msg *)
    While MsgUsed[MsgNum] <> 'F' do
        Begin
            MsgNum := MsgNum + 1;
            If MsgNum > MsgPerSec*NumSec then MsgNum := 1;
        End;
        MsgUsed[MsgNum] := 'T';
    End
Else
    MsgNum := MsgCount + StPtr;

ClrScr;
GotoXY (1,1); Write('Message :');
For i := 1 to 5 do
    Begin
        GotoXY (1,i+2);
        For j := 1 to 80 do Write (Msg[MsgNum,i,j]);
    End;
    If ForceQuest = 'Before' then GetTypAns(MsgNum);
    ScaleCount := 0;
    Repeat (* For all scales *)
        (* Pick Next Scale *)
        Scale := Ran (Seed,1,TotScales);
        While ScaleUsed[Scale] do

```

```

        Begin
            Scale := Scale + 1;
            If Scale > TotScales then Scale := 1;
            ..
        End;

ScaleUsed [Scale] := True;

(* Determine orientation and pointer position of display *)
Orientation := Ran (Seed,0,1);
PointerPos  := Ran (Seed,0,1);
DiffAns[MsgNum,Scale] :=
    GetAnsDisp (Scale,Orientation,PointerPos);

ScaleCount := ScaleCount + 1;

Until (ScaleCount = TotScales);

Until (MsgCount = MsgPerSec);

ClrScr;

If Section < NumSec then
    Begin
        GotoXY (1,1);
        Write ('Do you wish to go on to next section (Y/N) ? ');
        Read (Continue);
    End
Else
    Continue := 'N';

Until ((Section >= NumSec) or ((Continue = 'N') or (Continue = 'n')));
TempBool := False;

End;

ClrScr;

Quest := 0;

If DiffQuest = 'N' then Section := NumSec;

```

(* Determine If expert wishes to continue right now *)

If ((Section >= NumSec) and (ForceQuest = 'After') and
 (ForceDone <> 'Y')) then ..

Begin

 If Not (TempBool) then

 Begin

 Repeat

 GotoXY(1,1);

 Write('Would you like answer choice questions now ? ');

 Read (Kbd,Ch);

 Ch := UpCase (Ch);

 Until ((Ch = 'Y') or (Ch = 'N'));

 End;

 If Ch = 'N' then TempBool := True;

(* They want to continue so is Force question set for after all
 differential question. *)

If (Ch = 'Y') then

 Begin

 TempBool := False;

 ClrScr;

 For i := 1 to 120 do MsgUsed[i] := 'F';

 For k := 1 to NumSec * MsgPerSec do

 Begin

 GotoXY (1,1); Write('Message :');

 If Not (FixOrder = 'Y') then

```

Begin

    MsgNum := Ran(Seed,1,MsgPerSec*NumSec);

    (* Search for next available msg *)
    While MsgUsed[MsgNum] <> 'F' do
        Begin
            MsgNum := MsgNum + 1;
            If MsgNum > MsgPerSec*NumSec then MsgNum := 1;
        End;
        MsgUsed[MsgNum] := 'T';
    End

Else

    MsgNum := k;
    For i := 1 to 5 do
        Begin
            GotoXY (1,i+2);
            For j := 1 to 80 do Write (Msg[MsgNum,i,j]);
        End;
        GetTypAns(k);
    End;
    ClrScr;
    Ch := 'N';
    ForceDone := 'Y';
End;

End;

(* If forced question is after then find out if they now want to
. answer survey *)

```

```

If ((SurBool = 'Y') and (Section >= NumSec) and (SurDone <> 'Y')) then
  Begin
    If Not (TempBool) then
      Begin
        Repeat
          GotoXY(1,1);
          Write('Would you like to answer survey now ? ');
          Read (Kbd,Ch);
          Ch := UpCase (Ch);
          Until ((Ch = 'Y') or (Ch = 'N'));
        End;

        (* If there is a survey give it now *)
        If (Ch = 'Y') then
          Begin
            Assign (Survey,'Survey.Txt');
            Reset (Survey);
            Quest := 0;
            ClrScr;
            While Not Eof(Survey) Do
              Begin
                Readln (Survey,SurTyp);
                If SurTyp = 'Prompt' then
                  Begin
                    Readline;
                    GotoXY (1,1); Write (BlankLine);
                    GotoXY (1,1); WriteLine;

```



```

        Readline;

        GotoXY (1,2); Write (BlankLine);

        GotoXY (1,2); WriteLine;

        Readline;

        GotoXY (1,3); Write (BlankLine);

        GotoXY (1,3); WriteLine;

    End;

    If SurTyp = 'Reset' then
    Begin
        ClrScr;

        i := 0;

    End;

    If SurTyp = 'Question' then
    Begin
        Quest := Quest + 1;

        Readln (Survey,QuestType);

        Readline;

        i := i + 1;

        GotoXY (1,5+i); WriteLine;

        ReadLine;

        If Str80[1] <> ' ' then
        Begin
            i := i + 1;

            GotoXY (1,5+i); WriteLine;

        End;
    End;

```

```

ReadLine;

If Str80[1] <> ' ' then

  Begin
    i := i + 1;
    GotoXY (1,5+i); WriteLine;
  End;

XCursor := WhereX;
YCursor := WhereY;
if QuestType = 'X' then
  Begin
    Repeat
      GotoXY (XCursor,YCursor);
      Write (' ');
      GotoXY (XCursor,YCursor);
      QuestChar := ' ';
      Read (QuestChar);
      If QuestChar = #26 then QuestChar := ' ';

    Until QuestChar in ['X','x',' '];
    SurAns[Quest] := QuestChar;
  End;
if QuestType = 'Y/N/U' then
  Begin
    Repeat
      GotoXY (XCursor,YCursor);
      Write (' ');
      GotoXY (XCursor,YCursor);
      Read (QuestChar);

```

```

        Until QuestChar in
            ['Y','y','N','n','U','u'];
        SurAns[Quest] := QuestChar;
    End;
    if QuestType = 'F/C/U' then
        Begin
            Repeat
                GotoXY (XCursor,YCursor);
                Write (' ');
                GotoXY (XCursor,YCursor);
                Read (QuestChar);
            Until QuestChar in
                ['F','f','C','c','U','u'];
            SurAns[Quest] := QuestChar;
        End;
        if QuestType = 'Any' then Read (SurAns[Quest]);
        i := i + 1;
    End;
    Ch := 'D';
    End; (* End of Survey *)
    SurDone := 'Y';
    End; (* End of If Ch = 'Y' *)
    End; (* End of If section >= Numsec *)

If ((Section >= NumSec) and ((ForceDone = 'Y') or
    (ForceQuest <> 'After')) and ((SurDone = 'Y') or
    (SurBool = 'N')) then

```

```

    Begin
        GotoXY (1,3);
        Write ('All sections of this version of MsgJudge Complete.');
```

..

```

    End;

GotoXY (1,4); Write ('Please wait, writing Output Files...');

If ((Section < NumSec) or (Ch <> 'N') or (Ch <> 'n')) then
    Begin
        Rewrite (Status);

        Writeln (Status,Section:2);

        WriteLn (Status,ForceDone);

        Writeln (Status,SurDone);

        For i := 1 to MsgPerSec * NumSec do Writeln (Status,MsgUsed[i]);

    End;

If (Ch = 'D') then
    Erase (Status)

Else
    Close (Status);

WriteOutFile;

(* Rewrite confile to update the Seed *)

Rewrite (Confile);

Writeln (ConFile,Seed);

Writeln (ConFile,FixOrder);

Writeln (ConFile,DiffQuest);

Writeln (ConFile,SurBool);

Writeln (ConFile,ForceQuest);

```

```
WriteLn (ConFile,NumPageInst:3,NumSec:3,MsgPerSec:3,TotScales:3);  
WriteLn (ConFile,MsgType1);  
WriteLn (ConFile,MsgType2);  
Close (ConFile);  
End. 2
```

APPENDIX C

QUESTIONS ADMINISTERED TO PANEL

The following list of questions were presented to the subjects after they completed the MSGJUDGE program. These questions were computer administered.

Now that you have completed the self administered message judgment program, I would like to ask you some questions. I would like to remind you that your responses are confidential. Type 2-character answer.

What is your grade? (i.e. 01, 02, 03, 04, etc.)

What is your branch? (i.e. FA, IN, AR, AD, MI, EN, AV ,etc.)

Please answer with 'Y' for yes, 'N' for no, and 'U' for unknown/undecided.

Have you ever participated in a Command Post Exercise (CPX)?

Have you ever participated in a Field Training Exercise (FTX)?

I was asking you to rate messages that came from two different environments, CPX and FTX. Please enter 'Y' for yes, 'N' for no, and 'U' for unknown/undecided.

Do you think that it is possible, in general, to tell CPX messages from FTX messages?

What do you consider to be the recognizable differences in the two types of messages?

(MARK AS MANY AS APPROPRIATE WITH AN 'X',PRESS RETURN TO GO TO NEXT ITEM)

LENGTH OF TRANSMISSION?

VOCABULARY?

EMOTION OF THE SPEAKER?

SPEAKER, I.E. FIRST PERSON VICE SECOND OR THIRD PERSON?

COMPLETENESS OF SENTENCES?

ACCURACY OF DESCRIPTIONS?

PRESENT TENSE SPEAKER - VICE PAST TENSE SPEAKER?

USE OF KEY "TIP-OFF" WORDS?

Enter 'F' for FTX, 'C' for CPX, and 'U' for unknown/undecided.

Do you think that transmissions will be longer in an FTX or a CPX?

Will the speaker be more emotional in a CPX or in an FTX?

Will the speakers have a more varied vocabulary in an FTX or in a CPX?

Will the transmissions contain more accurate information if from a CPX or from an FTX?

Enter 'Y' for yes, 'N' for no, and 'U' for unknown/undecided.

Do you feel that the quality of training could be affected based on the radio transmissions that describe the battlefield to the commander and staff?

Have you ever participated in an opposing force exercise at the National Training Center?

Have you ever participated in a CPX using ARTBASS?

Enter 'C' for CPX, 'F' for FTX, or 'U' for Unknown/Undecided.

Which environment, FTX or CPX, provides the best training for yourself?

Which environment, FTX or CPX, provides the best training for the battalion staff?

The self administered program you just completed is the initial phase of a project to describe differences in communications between a CPX and an FTX. The goal of the project is to provide more realistic training environments. Do you think that this self administered message judgment program will show those differences? Enter 'Y' for yes, 'N' for no, or 'U' for unknown/undecided.

I also wish to determine if there is a training difference between the CPX and the FTX. I am going to use the differences in message traffic to describe that difference. Enter Y for yes, N for no, U for unknown/undecided.

Do you think that this program will help to identify the training differences?

Is this type of research, determining differences between CPX and FTX messages to improve training, beneficial to the Army?

THANK YOU FOR YOUR TIME AND YOUR RESPONSES.

APPENDIX D

MESSAGES USED IN MSGJUDGE

Transmissions: TFCMD EE

..

1. Roger, time was 0514. Three tracked vehicles, one confirmed. Grid 246 070. It's a small reconnaissance patrol moving slowly North East.
2. T62 tank destroyed, 560 126.
3. Roger, we got one T72 and one BMP, grid 531 055.
4. Incoming edbirds.
5. Roger, I know they aren't supposed to be using them. My guys reported a motorcycle, grid 585 125.
6. Can you pass onto Whiskey, that motorcycle, the last we could tell it looked like it was going into that high ground. Probably in his area.
7. Roger, large levels of smoke building, grid 430 110.
8. All stations this net, all stations this net, Redbird, Redbird. Coming West to East.
9. I've got an OPFOR forward command post, grid 497 095. He's sitting up there now and calling things in.
10. 4 BMPs, 261 789.
11. Roger, we need artillery on that grid I sent you. You have an OPFOR commander sitting there on top of a hill. All you have to do is blow him away with artillery. He's got one jeep with two antennas sticking up.
12. Spot report. BMP with dismounted troops, 083 765.
13. BMPs, vicinity 54 08. Might make contact across the front.
14. Grid 567 102, looks like three PCs.
15. Observing smoke, and artillery grid 575 112.
16. We have 21 Victors coming out of the wheel gap at this time.
17. Observing artillery, grid 585 170. Continuing mission.
18. Artillery, grid 582 161. Enemy artillery, grid 582 161.

19. 33 vehicles, grid 487 029; 28 vehicles, grid 519 020; 51 Victors, one MRB moving NorthEast. No grid available. Giving indirect fire at this time.
20. Four identified columns. Lead element of first column 486 023. Approximately 35 Victors. Second column 545 025. Heading north, also. Approximately 35 Victors. Third column vicinity the whale gap is stationary, vicinity 485 987 heading North, correction, stationary column. Final column, 516 024, approximately 30 vehicles heading North.
21. Smoke, grid 595 986. Moving SouthWest.
22. Gas, Gas. Bravo Papa 60.
23. We have approximately 100 Victors to the North. Continuing to engage.
24. A10 aircraft says 15 T72s proceeding North through red pass on the road.
25. Observing artillery, grid 592 150. Continuing mission.
26. Redbird, East to West, 444 143. Gas and spray.
27. It appears that they have not penetrated, but they are really just beginning to get to their front trace.
28. Roger, moving East, 3 T72s. Grid 589 116.
29. They have a total of 11 tracked vehicles vicinity 475 985. They are beginning to head North along trail.
30. Destroyed one T62, 565 120. Engaged other elements: Either we destroyed them or someone else did. Grid 575 125. No losses to friendlies. Observing force moving out of red pass towards Uniform, and toward Bravo.
31. We've got green smoke in vicinity grid 585 105. Killed 4 BMP at 345 561.
32. Roger, we have a penetration. The force is moving in the vicinity of 581 720. We've had friendly air going in on that penetration. Firing with artillery. We are retaining our position in 60 and 61 and 62 and firing into engagement areas.
33. I can confirm one 122, 4735 9835.
34. Roger, they're down in the gap.
35. Roger, there is a group of vehicles still there. There are 11 vehicles vicinity 523 069. They are hiding in the wadi by those fingers.
36. Vicinity 523 069. The fingers jutting out and they are hiding in the wadi. In that vicinity.

37. Roger, 15 vehicles moving NorthEast. From vicinity 520 110.
38. Roger, it appeared like a MRB was coming into the flank of 60 on to your other element, over there.
39. They're holding their position. They don't have any enemy in sight, that is still alive.
40. I want to roll to 60. Move to Bravo Papa 60. We are being penetrated in the rear. You have to bring what you've got right into 60.
41. Pass on to higher that we are beginning to be penetrated. Another, appear an MRB or reinforced, almost ARMB, is beginning to penetrate 80. I'm not sure we can hold DOG. I'm counter attacking into the flank.
42. We are beginning to be penetrated in our northern most position. I'm counter attacking in flank of the OPFOR.
43. Roger, I'd estimate about an MRB beginning to penetrate. I'm counter attacking out of 61 into his flank, but he's rolling right through 60 right now.
44. Roger, my element at 60 attempting to fight, but there's too many of them.
45. Roger, I anticipate a small penetration. Right now we are counter attacking. I will keep him informed.
46. Roger. It is not a large penetration, but there is approximately, I'd say 2 T72s, a couple of BMPs. I do have an element somewhat same size maneuvering against it now.
47. Stay off the net a second. Most of the forces are East of me. It looks like they are bypassing us. I'm moving mine in a counter attack right now against them. If you look for the smoke you can see where the battlefield is. If you've got anything left, aim it in that direction. They should be coming in your range soon.
48. I'm in 60 trying to force out whatever got through. Got Redbirds coming in this area. Redbirds, Redbirds.
49. Roger, we're taken down. There's only one left. He's headed for Uniform. One T72 and one BMP left.
50. 16 tracked vehicles. BMPs, 492 032, moving slowly towards NorthEast.
51. 492 032. 16 vehicles moving slowly.
52. I now have 02 of line 1, item 1 left. Had a close in fight with what we thought was a dead BMP. He was just playing possum. Killed three of mine.

53. I don't see anything out here. I don't see anything out here but junk Victors.

TFCMD RO

1. Roger, they are doing it now. Moving to our west.
2. Roger, friendly force moving off to our West, on main East/West.
3. He is currently engaged. Will contact you as soon as possible.
4. They are conducting a mission to our West. They have already moved past my force.
5. Friendly force has passed through our area.
6. Friendly force has passed through our area. Request that you relay through me, to him, whether passage of lines was coordinated.
7. We have already confirmed with force 7 Charlie that 24 is on the way. The three tracks, they stopped 056 014. Had a relay net going indirect on that.
8. I'm in the Southern part of 61.
9. As far as I know, they have passed.
10. Mike 08 says that as far as he knows all friendly units have passed the lane. Requesting permission to close at this time. I was trying to get ahold of 47 to see if D47 has contacted him, whether or not all friendly units have passed.
11. They're stationary at this time.
12. Roger, request is in. We should know something in 5 to 10 Mikes.
13. We're moving.
14. Affirmative, they are kind of wandering through my area. Trying to run them down. Do you know where they are supposed to be?
15. Negative, we didn't have any of those. We had some Charlies.
16. Already doing so.
17. I'd like same position you just left.
18. Well, your one's enroute.
19. We finished getting them all dug in last night.
20. They're staying back. I'm not going to not going to use them. To be prepared to move on a counter attack, all my heavy assets are in, or prepared to move.

21. Roger, I was told way back when, we had one on station at 0630. What happened to that?
22. Negative. We were told we would have one at 0630. I understand it has just departed. ..
23. Bluebird, Bluebird, moving NorthEast to SouthWest. Bluebird, Bluebird, NorthEast to SouthWest.
24. Observing artillery, grid 585 170. Continuing mission.
25. Roger, the same as before. We have the same front line trace.
26. Observing friendly artillery battery. From Delta 25 down 5.2, right 1.5. Thus endangering. If we could move it around the west side.
27. I say again, observing friendly artillery battery. From Delta 25 down 5.2, right 1.5. They are not interfering with my fire, but they will certainly become a lucrative target if they stay there.
28. 33 vehicles, grid 487 029, 28 vehicles, grid 519 020. 51 Victors, 1 MRB moving NorthEast. No grid available. Giving indirect fire at this time.
29. On their way.
30. Am engaged in sector.
31. Roger, we are diverting.
32. Observing artillery, grid 592 150. Continuing mission.
33. If Quebec or Uniform have any elements which can observe, can observe yellow smoke, they'll be able to see the OPFOR.
34. 4 A-10s inbound.
35. Destroyed one T62, 565 120. Engaged other elements. Either we destroyed them or someone else did. Grid 575 125. No losses to friendlies. Observing force moving out of red pass towards Uniform and toward Bravo.
36. We've got green smoke in vicinity grid 585 105.
37. Roger, we have a penetration. The force is moving in the vicinity of 581 720. We've had friendly air going in on that penetration. Firing with artillery. We are retaining our position in 60 and 61 and 62 and firing into engagement areas.
38. Orienting my force here. Due South to engage.
39. I'm on that already. I'm here in force. I'm standing by and where you wanted me to be, now.

40. Roger, we got elements both that way and the other side.
41. Roger, I'm to go on yellow 5, line Alpha Bravo. We're bring some up from Charlie at this time. I might not have enough to do much damage to this echelon coming in now. ..
42. I appear to have stopped the first echelon. This position, I have 03 line 1 left. Another position, I have 04 line 1 left. My vehicle has been shot out from under me.
43. I want to roll to 60, move to Bravo Papa 60. We are being penetrated in the rear. You have to bring what you've got right into 60.
44. Pass on to higher that we are beginning to be penetrated. Another, appear an MRB or reinforced, almost ARMB, is beginning to penetrate 80. I'm not sure we can hold DOG. I'm counter attacking into the flank.
45. We are beginning to be penetrated, in our Northern most position. I'm counter attacking in flank of the OPFOR.
46. My element in 60 North had nothing to shoot at. I'm moving him to get the enemy in the rear.
47. Roger, I'd estimate about an MRB beginning to penetrate. I'm counter attacking out of 61 into his flank, but he's rolling right through 60 right now.
48. Roger, my element at 60 attempting to fight, but there's too many of them.
49. Roger, I've got an element almost up to full strength in 59. He's counter attacking into the North, into the flank.
50. Roger, I'm rolling around the back end of 60 now. Up on the rear of what's left of that LRC.
51. Roger, I anticipate a small penetration. Right now we are counter attacking. I will keep him informed.
52. Roger. It is not a large penetration, but there is approximately, I'd say 2 T72s, a couple of BMPs. I do have an element somewhat same size maneuvering against it now.
53. Stay off the net a second. Most of the forces are East of me. It looks like they are bypassing us. I'm moving mine in a counter attack right now against them. If you look for the smoke you can see where the battlefield is. If you've got anything left, aim it in that direction. They should be coming in your range soon.
54. I'm in 60 trying to force out whatever got through. Got Redbirds coming in this area. Redbirds, Redbirds.

55. I've got my November 71 moving in from the due West. He'll be here in about 10. Got a few elements left of my other element. They're getting, they're re-consolidating their ammunition so they can continue to fight. Go ahead and identify the enemy force. We'll assist you.
56. Roger, that's me and what I got left that rolls.
57. I'm coming that way. I'm headed right for your own position. I'll reinforce you and meet you there.
58. I should be there in about 10 Mikes.
59. Roger, we're taken down. There's only one left. He's headed for Uniform. One T72 and one BMP left.
60. We're firing indirect now.
61. I now have 02 line 1, item 1 left. Had a close in fight with what we thought was a dead BMP. He was just playing possum. Killed three of mine.
62. Line 1, item 1, 02. Holding position with what's left. Distributing ammunition among what vehicles are remaining alive.
63. Roger. Line 1, item 1, 01; which is mechanical failure still in position, but can still fight. Line 1, item 2; 03, 05, 02, 04, and 55. I've swept them to the border. They've gone through the DS element there. I've taken what I've got left and moved back to my position. I've still got elements there dug in who can fight, and link up with the Uniform element and beef him up.
64. Line 1, 0, 3; Line 1, 00 , 03, 02, 06; 69 percent.
65. Roger, I'm moving back from what I had, and re-occupying 61.
66. Did you get through your friendly obstacle? I need somebody out here to guide me through?
67. Roger, he has already established that and he is working that. He is with higher on that and is coordinating with the appropriate individuals to insure a contiguous move.
68. We've considerable reconstitution to do.
69. My element is currently at the link up point. At the fire hydrant. Should commence operations in about 15. The OPFOR got through with a total of 7 vehicles, and one, that's what, that's what made it past us. us. We saw them go by. That's what made it to the boundary. One jet mission would have taken them out.
70. Roger, I think Bravo would have policed up that with what he was doing.

71. I'm in Bravo Papa 65 with green smoke in hand.
72. Affirmative. I'm dirty.
73. I'm clean. I can't come over there. I have been on the horn to Yankee, talking to their 37, their 22. The party had a mission to come 1 kilometer and linkup. He's lost and he's looking for them. Since I can't get in there, how about you getting out and sorting this out. And seeing if there's a representative from Yankee, in order to begin operations.

BNCMD EE

1. We spotted three more batteries moving, correction, three BMPs moving East from 327 899. Have exchanged fire. Believe two BMPs destroyed.
2. I got 4 BMP vicinity 360 895. Two confirmed kills at 323 289.
3. Roger, 4 BMPs element is at South 1.8, West 1.8 from A 04.
4. Roger, one more confirmed kill on BMP at 372 891.
5. Roger, I have one more confirmed BMP kill. At 372 891.
6. Spot report. Spotted three tracked vehicles, vicinity 359 895. They are now in direct fire with my element.
7. We need fire on 3388 2954. BMPs moving East. I'm observing smoke at that location.
8. Got personnel to my front.
9. Spot report. Alpha spots 5 T64 at grid 362 890, moving east.
10. Spot report. T&E detected 7 BMPs total. 3 BMPs moving East, 272 961. 4 BMPs moving east from 276 966.
11. Roger, T&E detects 7 BMPs total. 3 BMPs at grid 272 961. 4 BMPs, grid 276 966, time 0128.
12. I've got visual detection, 5 T64s. Vicinity 361 906, has returned fired. Have one confirmed kill.
13. Roger, they are moving at this time. Have one BMP located at 357 901, moving West.
14. I've got these BMPs and a tank platoon it seems down in the southern sector. I've got seven BMPs in the North. What are we facing?
15. There are only three BMPs in vicinity 362 923.
16. Two T64s, one BMP moving East.

17. Spot report. OP 5 detects 1 BMP moving East on the road, grid 276 907. Also, OP 3 has taken a lot of direct fire. Reports that his lines are down. He has taken a lot of that fire through friendly illumination.
18. OP 3 is still alive. He now has detected 7 personnel moving northeast from 364 902.
19. Southern element has detected seven personnel moving east at 364 908.
20. Spot report. Last BDRM now at 300 913, continues to move East. Request artillery.
21. Visually detected 2 T64s, grid 371 918.
22. Am engaging 2 T64s at grid 372 917.
23. I have 4 friendly Victors, one 64 to my front.
24. Have killed one T64 in vicinity 371 919.
25. G1B has detected 1 light track vehicle activity unknown at grid 310 968.
26. Roger, visually detected 1 BDRM, 1 73mm BMP moving East, from grid 323 967.
27. Southern element is under fire at this time. Have lost one Victor and 6 personnel.
28. Negative, personnel are dismounted.
29. Have destroyed the following: 1 73mm BMP, vicinity 348 966.
30. OP 5 has visually detected 1 BDRM, 1 73mm BMP at grid 276 907 moving East on the road. OP 1 has detected one BDRM moving northeast on the road, at 301 931.
31. Roger, be advised that northern element is taking illum at this time.
32. Roger, I need artillery at 290 909. 1 BMP, 1 BDRM, stationary.
33. 1 BMP destroyed. Still have enemy dismounted moving in the vicinity of 288 912.
34. OP 1 spots 1 BDRM, 1 BMP moving northeast on the road from 291 933. OP 1 is receiving direct fire, also. 1B has detected 3 light tracked vehicle, activity unknown, at grid 303 970.
35. Scout 2 has just received 1 BMP and 3 truck. Believed to be engineers, moving East, from 289 971.
36. Spot report from Scout 1, an unknown enemy spying on him from 311 899.

37. Roger, I have him spotted. 1 BMP, grid 305 957.
38. Roger. He's at grid 321, correction, 330 975. Also GSR in the North GlA, correction, Bravo, is under suppression from a force at 342 978.
39. G2A has detected a light tracked vehicle with radar at 307 957.
40. Have killed 1 BDRM vicinity 366 919, with center element.
41. Roger, Have destroyed one more BMP at vicinity 366 920.
42. Have destroyed 1 APC vicinity 366 920. Have visually detected 9 personnel at 336 956. Northern element has also detected 9 personnel at same location.
43. Spot report. Observe 9 personnel at grid 336 956. They are moving northeast.
44. Headquarters element has destroyed 1 BDRM vicinity 358 964.
45. I believe they're trying to come North, due to engineer support in the North.
46. Have destroyed 2 BDRMs vicinity 358 964.
47. OP 51 and 2, have negative sit rep left of their location. Scout 5 just gave me spot report of 15 personnel at 355 978. Activities unknown, will verify.
48. Have visually detected 1 BDRM vicinity 370 942. Two dismounted moving East at 370 914.
49. Roger, same element has destroyed 1 BDRM located 388 927.
50. Roger. Be advised Scout 5 and 3 have taken heavy fire. OP 4 confirms that he was destroyed.
51. Visually detected 3 BMPs location 440 893.
52. Spot report: 3 BMPs location 447 894, moving East.
53. Dismounted activity? The only one left, last spot report I had, was vicinity of 377 910. Do you have something that I don't?
54. I have visually detected 2 BMPs, vicinity 448 895.
55. I suspect enemy first echelon is reaching the Hotel phase line on my time phase line. For planning purposes, I put them there in about 1 to 2 minutes.
56. Have destroyed 3 BMPs, location 454 925.
57. Flags, Flags, Flags, Bandits, Bandits 25 95. I say again 25 95.

58. Detected 3 ZIL 131 trucks grid 389 901.
59. We spotted 10 BMPs, at least 18 T64s and 3 trucks moving East from grid 296 971. Lead elements of that force are in contact with OP 2.
60. We have detected 2 BMP at 256 947. ..
61. Direction of flight is West to East.
62. Artillery has ceased at this time, and no casualties taken.
63. Receiving light suppression.
64. We have 4 Hinds. The first 2 HIND Ds are in vicinity 385 902. The second 2 HIND Ds are 355 902.
65. Bandits, bandits, bandits. 32 90 and 35 90, HIND Ds. Acknowledge.
66. Flash, bandit warning, high 32 90 and 35 90, flying East.
67. Roger, taking fire at this time, taking fire at this time.
68. Those HIND Ds blew away all 4 tanks in 1 platoon.
69. Element at 466 930 under heavy air attack.
70. Bandits flying West to East.
71. We've detected 3 BMPs, 5 T64s, a BDRM, and 3 truck vicinity 37, correction, 366 960.
72. OP 4 has detected 3 BMPs, 5 T64s, 3 trucks at grid 366 960.
73. Spot report. 3 trucks, 3 trucks moving East vicinity 416 893, also BMPs several clicks behind at 388 959. Moving toward chemical area.
74. Receiving artillery at this time.
75. Spot report: 3 BMPs moving East grid 394 968. Also 2 more BMPs moving East grid 285 957. Taking fire at this time.
76. A platoon of BMPs located 43 91. Platoon of tanks 37 90 and, company plus of BMPs plus 2 platoons of tanks at 35 91.
77. 1 platoon tanks 37 90.
78. Combat status 5465. That engagement that was going just South of me, dismounted personnel, approximately 3 or 4 right now.
79. Visually detected 4 T64s, location 418 902. 418 902, moving West, Roger East.
80. Receiving indirect fire on position. South 1, West 2, on A 04, correction A 01.

BNCMD RO

1. Position update: Time 0026, one element at, from A 03 West 3, correction, West 2.5, North 1.5. Second element from A 03 South 2.5, West 1.0. Last element from A 04 South 3.0, West .5.
2. Roger, location from one section from A 03 West, correction, East 1.5. Second section from A 03 South 2.5, West, correction, East 1.0.
3. We spotted three more batteries moving, correction three BMPs moving East from 327 899 have exchanged fire. Believe two BMPs destroyed.
4. We have the N09 element in position, and if you call for fire on that we can help you out.
5. We have lost contact with that element.
6. Roger, element is South 1.4, West 1.4.
7. Roger, they are in position right now. West 2, North .5.
8. 6 kills.
9. Spot report, spotted three tracked vehicles vicinity 359 895. They are now in direct fire with my element.
10. Roger. I'm taking casualties at this time.
11. 94 minus 2.
12. I have lost night vision at my southern element. Can't swing by with central element.
13. Also, OP E has not reported in since that time.
14. Current location 2.2 South, 1.7 West of A 04. Receiving indirect fire, rear of position.
15. I've got visual detection, 5 T64s, vicinity 361 906, has returned fire. Have one confirmed kill.
16. Roger, I'm having problems on hill 95 dropping illum.
17. Spot report OP 5 detects 1 BMP moving East on the road, grid 276 907. Also, has taken a lot of direct fire. Reports that his lines are down. He has taken a lot of that fire through friendly illumination.
18. Negative, due to direct fire from those BRDMs and tanks, East of his location. But they spotted him due to friendly illumination.
19. OP 3 is still alive. He now has detected 7 personnel moving northeast from 364 902.

20. Am engaging 2 T64s at grid 372 917.
21. I have 4 friendly Victors, one 64 to my front.
22. Have killed one T64 in vicinity 371 919.
23. Southern element is under fire at this time. Have lost one Victor and 6 personnel.
24. Have destroyed the following: 1 73mm BMP, vicinity 348 966.
25. Roger, be advised that northern element is taking illum at this time.
26. Have taken 2 more personnel casualties. Lost 1 NOD.
27. 80 percent.
28. Taking losses. Lost 1 M2, northern sector.
29. I have moved southern element, southern 1.5 West, correction, East .5, from A 02.
30. 1 BMP destroyed. Still have enemy dismounted moving in the vicinity of 288 912.
31. OP 1 spots, 1 BDRM, 1 BMP moving northeast on the road from 291 933. OP 1 is receiving direct fire, also. 1B has detected 3 light tracked vehicle, activity unknown, at grid 303 970.
32. Northern element is moving .01 North, .01 East of A 03. Center sector is moving .8 West, .01 North of A 04.
33. Northern element will be at A 03.
34. Am in the amber status on fuel. Approaching red status on ammunition on all sectors.
35. I have an illumination approximately 4 clicks behind my center location at this time.
36. Roger, I have elements ready to engage at this time.
37. Roger. He's at grid 321, correction, 330 975. Also GSR in the north GlA, correction, Bravo, is under suppression from a force at 342 978.
38. Have killed 1 BDRM vicinity 366 919, with center element.
39. Roger, Have destroyed one more BMP at vicinity 366 920.
40. Center sector taking heavy fire at this time.
41. Headquarters element has destroyed 1 BDRM vicinity 358 964.

42. Strength is 2 vehicles. Cannot engage with center sector. Pulled back too far.
43. 2 M3s, 6 OPs.
44. Both radar up.
45. Have destroyed 2 BDRMs vicinity 358 964.
46. It's moving into position at this time.
47. Approximately 80 percent.
48. With no illumination, I am totally useless in the South. Any NODS on any southern element, at this time?
49. Status: 5 M3 zapped, 1 OP zapped, both GSRs dead.
50. Roger. Our status as of 0415: 1 M3 good, 1 OP zapped, 2 GSRs dead, 5 M3 zapped.
51. Status is approximately 70 percent. Returning to rear.
52. 1 in the northern, 2 in the southern.
53. I have your tank in sight. I'm moving back in at this time.
54. OP 51 and 2, have negative sit rep left of there location. Scout 5 just gave me spot report of 15 personnel at 355 978. Activities unknown, will verify.
55. It's useless out here.
56. Juliet has effected a linkup and is moving at this time.
57. We have affected linkup and moving back into perimeter.
58. Roger, same element has destroyed 1 BDRM located 388 927.
59. Receiving friendly artillery. I have right now lost seven people and am under heavy suppression. I'm at A 02 West 2.
60. Roger. Be advised Scout 5 and 3 have taken heavy fire. OP 4 confirms that he was destroyed.
61. Moving into position at this time.
62. Receiving incoming at this time.
63. Roger, going to MOPP 4 at this time.
64. Roger, reporting still receiving incoming artillery.

65. Dismounted activity? The only one left, last spot report I had, was vicinity of 377 910. Do you have something that I don't?
66. South 4, West 1, from A 04.
67. Correction on that. South 2.4, West .5.
68. We spotted 10 BMPs, at least 18 T64s and 3 trucks moving East from grid 296 971. Lead elements of that force are in contact with OP 2.
69. Artillery has ceased at this time, and no casualties taken.
70. Receiving light suppression.
71. Combat status 100 percent.
72. Roger, taking fire at this time, taking fire at this time.
73. Those HIND Ds blew away all 4 tanks in 1 platoon.
74. Have a platoon tank, 1 platoon size 100 percent, 1 platoon tank totally wiped out.
75. Element at 466 930 under heavy air attack.
76. Combat received 1 tank 3 Bradleys.
77. Receiving artillery at this time.
78. Alpha element has released attack helicopter company to us. Call sign 66. Will meet us at wheel gap approximate 15 Mikes.
79. Am at same location -break- fire has shifted, as a matter of fact, it has ceased. Combat status is 94 108.
80. Spot report: 3 BMPs moving East grid 394 968. Also 2 more BMPs moving East grid 285 957. Taking fire at this time.
81. Combat status: 1 tank left, 1 tank left, and moving North to the, to better ground at this time.
82. Status is 1 ITV, 1 M2 remaining.
83. Combat status 5465. That engagement that was going just South of me, dismounted personnel, approximately 3 or 4 right now.
84. Status, is 1 slant 0 slant 4.
85. Receiving indirect fire on position. South 1, West 2, on A 04, correction A 01.

86. Current combat status is 1 ITV, 1M2. Current location above A 02 West 1, South 1.
87. Roger. Be advised Scout 5 and 3 have taken heavy fire. OP 4 confirms that he was destroyed.
- ..

APPENDIX E

INSTRUCTIONS TO PANEL

Below are the instructions that each subject read prior to the MSGJUDGE task. They were displayed on the computer screen. The "more..." indicates that there are additional parts to read.

INTRODUCTION

Welcome to MsgJudge. The task you are to perform is one of rating radio messages; it is not an exam with right or wrong answers. I am interested in your impressions, and ask that you work quickly. Your answers will be used for research purposes only and will be confidential.

SETTING--Put yourself in the role of a task force commander and two of his principal staff members, the S-2 and the S-3. Your battalion is conducting a combined arms operation and you have been given the mission of defending a portion of the battlefield.

THE TASK--You will be presented with a series of messages and asked to rate the messages on a bi-polar scale. These messages have been transcribed from actual radio traffic recorded during training exercises. When you rate the messages, think of the environment that has been set for you and ask yourself about the reality and utility of the message. Your rating should be geared to asking what kind of action you would or could take based on the subject message. Although the messages are not presented in a chronological or contextual format, I feel that you will be able to rate each message in this manner.

more...

PROGRAM INSTRUCTIONS

This program is presented in three major parts. At the end of each of these parts and at certain intervals during Part 1, the user is allowed to exit MsgJudge and return at a later date with no loss of data. Part 2 and 3 will require less time to complete than Part 1.

Part 1 will display a message and a scale from -3.5 to 3.5, with differential adjectives at either end. The user should note that the orientation of this scale will not always be the same. To enter your answer place the pointer in desired position and strike the return key on the keyboard. To move the pointer use the less than/more than (< / >) keys to move one position at a time, or press shift key and less than/more than to move 5 positions at a time.

In Part 2 and 3, enter your answer and press the return key.

End of instructions...

APPENDIX F

SOURCE TABLE FROM ANALYSIS

Below are the source tables from the preliminary analysis results for the scales and forced choice by subjects.

TABLE F-1

Subject Effects on the Semantic Differential (SD) Scales

Independent var: Subject						
Dependent var: SD						
NUMBER OF CASES PROCESSED: 180						
DEPENDENT VARIABLE MEANS:						
SD(1)	SD(2)	SD(3)	SD(4)	SD(5)	SD(6)	SD(7)
0.921	0.356	0.608	1.057	0.471	1.041	0.562
SOURCE	SS	DF	MS	F	P	
Subject	1.292	2	0.646	0.099	0.906	
ERROR	1154.549	177	6.523			
SD	87.540	6	14.590	10.538	0.000	
ERROR	1470.346	1062	1.385			
Subject X SD	46.689	12	3.891	2.810	0.001	
ERROR	1470.346	1062	1.385			

TABLE F-2

Subject Effects on the Forced Choice (FC) Questions

Independent var: Subject					
Dependent var: FC					
N: 180					
MULTIPLE R: .047					
SQUARED MULTIPLE R: .002					
ANALYSIS OF VARIANCE					
SOURCE	SS	DF	MS	F	P
SUBJECT	0.100	2	0.050	0.199	0.820
ERROR	44.450	177	0.251		

APPENDIX G

PRELIMINARY ANALYSIS RESULTS

The results of the preliminary Analysis for the scales by message source and content are given below.

TABLE G-1

Source and Content Effects on the Semantic Differential (SD) Scales

Independent var's: Source, Content

Dependent var: SD

NUMBER OF CASES PROCESSED: 180

DEPENDENT VARIABLE MEANS:

DIFFS(1)	DIFFS(2)	DIFFS(3)	DIFFS(4)	DIFFS(5)	DIFFS(6)	DIFFS(7)
0.921	0.356	0.608	1.053	0.471	1.041	0.562

TEST OF HYPOTHESIS

SOURCE	SS	DF	MS	F	P
Source	18.579	1	18.579	3.303	0.071
Content	147.156	1	147.156	26.161	0.000
Source X Content	0.111	1	0.111	0.020	0.889
ERROR	989.997	176	5.625		
SD	87.540	6	14.590	10.739	0.000
SD X Source	12.163	6	2.027	1.492	0.177
SD X Content	45.384	6	7.564	5.568	0.000
SD X Source X Content	24.879	6	4.146	3.052	0.006
ERROR	1434.609	1056	1.359		